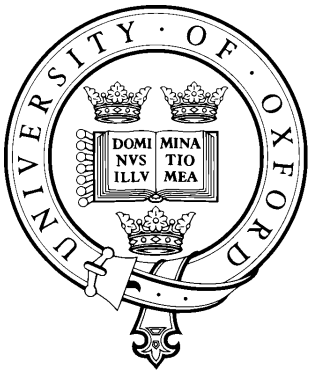
Security Test and Evaluation of Cross Domain Systems

Joe Loughry

St Cross College

University of Oxford



A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas Term 2014

Build 4642

Abstract

In practicable multi-level secure systems it is necessary occasionally to transfer information in violation of security policy. Machines for doing this reliably and securely are called cross domain solutions; systems incorporating them are cross domain systems. Data owners, especially in classified environments, tend to distrust other data owners, other systems and networks, their own users, and developers of cross domain solutions. Hence, data owners demand rigorous testing before they will allow their information into a cross domain system. The interests of data owners are represented by certifiers and accreditors, who test newly developed cross domain solutions and newly installed cross domain systems, respectively. Accreditors have the authority to grant approval to operate and the responsibility for accepting residual risk. Certification and accreditation have always been expensive and time consuming, but there are hidden inefficiencies and unexploited opportunities to predict the actions of accreditors and to control the cost of certification. Some case studies of successful and unsuccessful security certifications and accreditations were analysed using grounded theory methodology. It was discovered that inefficiency arises from conflation of the principle of defence in depth with the practice of independent verification and validation, resulting in an irresistible appearance of cost savings to managers with a possible explanation in the relative maturity of different levels of software engineering organisations with respect to policy, process, and procedures. It was discovered that there is a simple rule relating certifier findings to developer responses that predicts the duration of penetration testing and can be used to bound the schedule. An abstract model of cross domain system accreditation was developed that is sufficiently powerful to reason about collateral, compartmented, and international installations. It was discovered that the behaviour of accreditors satisfies the criteria for reliable signalling in the presence of asymmetric information due to Akerlof, Spence, and Stiglitz.

Acknowedgements

To my supervisors, Dr Andrew Martin and Dr Ivan Fléchais, for their help and guidance and for starting the Oxford Security Reading Group. To Lt. Col. Tom Flaherty, U.S.M.C. (ret.), who hired me into the closed area. And to my family.

Contents

List of Tables

List of Figures

Preface

This research was made possible in part by the U.S. Air Force Research Laboratory (AFRL) under contract number FA8750-09-C-0006 issued by AFRL/RIEB. The author wishes to express grateful appreciation to the Lockheed Martin Corporation of Bethesda, Maryland for access to historical records and data.

Chapter 1

Introduction

Sometimes it is necessary to violate your own security policy.[1] The necessity for this presents itself from time to time in multi-level operating systems that use Mandatory Access Control (MAC) because multi-level systems employing MAC are well suited to keeping information—and users at different security classifications—separate, but in practice sometimes systems need to interact and share differently classified information in order to make use of it [74, 228, 267]. There are a handful of recognised solutions to the problem of controlled communication across security boundaries; they may be designed in, as between partitions in a separation kernel operating system; or external, as in the case of routing devices or protocols between compartmented mode workstations over a single-level network or multi-level network, or between computers on isolated networks in different security enclaves. While unneeded in certain special applications of multi-level systems—*e.g.*, in consolidated aircraft flight control computers where different safety-critical programmes run in separate partitions on the same hardware in order to conserve space, weight, and power but never communicate or share

[1] Some of the information in this chapter has been previously published in the *C&ESAR 2012* conference, Rennes, France, pp. 19–28 [143].

data across partition boundaries [197]—nevertheless in numerous other cases including imagery analysis and intelligence sharing, communication across security boundaries in multi-level systems is a requirement. When recently a new high-assurance separation kernel was developed for certification, an inter-partition communication subsystem had to be developed and certified at the same time [3, 104, 206, 248, 249].

1.1 Subject of this Thesis

The general subject of this thesis is problems encountered by developers, certifiers, and accreditors when attempting to evaluate and certify the security of systems, like that one, that need to communicate across security boundaries. More particularly, what happens when the security testing criteria are new or have suddenly changed? Most specifically, is the high cost of security testing a necessary consequence of information assurance, or are there ways to make it quicker without making it weaker?

1.1.1 Method

To answer that question, we look at a longitudinal study of software engineers, testers, certification authorities, and accreditors (who represent the interests of data owners) of a typical cross domain system used for classified information. Case studies exhibiting both successful and unsuccessful outcomes of security testing are fairly presented. From these, we derive a grounded theory of formal and informal communication channels that exist in the certification and accreditation process and which may explain why some of these characteristic and puzzling behaviours persist. To assist with understanding projects that are done in classified environments and take years to complete at a cost of hundreds of thousands or millions of pounds,

an abstract model of certifier and accreditor behaviour—one that satisfies the conditions given by Akerlof and Stigler for 'reliable signalling' in the presence of asymmetric knowledge—was developed.

1.1.2 Previous Publications

All of the methodology and results herein have been previously published in preliminary form. The conflation of the principle of defence in depth with the practice of independent verification and validation, together with overlapping domains of responsibility leading to unnecessarily high cost of cross domain certification and accreditation, together with a research plan including only the failed Common Criteria security evaluation—what is now Chapter 4—was published in 2010 [142]. With the addition of another case study extending the research longitudinally across the evolution of an exemplar cross domain solution (Chapter 5), the residual risk calculation between accreditors on different sides of a multi-level system, the existence of at least one covert channel, and the forcing of undesirable information flows (Chapter 6), were published in 2012 [143]. The abstract model was then extended to include the accreditation phase in addition to certification testing, and published along with new results in 2013 [144].

1.1.3 Applicability

Despite its origin in intelligence community and military communication systems ranging from radar tracking to combat search and rescue to signals interception, this research may find its first practical application in the field of health care. Consider the problems of health insurance, moral hazard, and adverse selection. Health insurance companies, it may be observed, act a bit like competing foreign intelligence agencies: it is not far off the mark, if you

take into account *all* of their motivations. The problem of privacy-respecting Electronic Health Record (EHR) systems and information sharing between health care providers and insurance companies reduces to an SCI-like or international cross domain system accreditation with non-cooperating data owners (see Chapter 6).

1.2 Crossing Domains

The reason for existence of cross domain systems is to violate security policy in a controlled manner [5, 74]. In practice, despite the existence of published standards and commercially available operating systems, at the present time it must be admitted that compartmented mode workstations and multi-level servers are comparatively rare in the wild. Instead, in the existing Intelligence Community (IC) environment, made up as it is primarily of single-level networks—that is, relatively isolated networks each of which comprises a security enclave or security domain at a particular security classification—connected to the network cloud, the prevailing security policy for military and national security systems remains Bell–LaPadula (with the addition of Biba's '$*$'-integrity axiom in some cases)[2] [21, 25]. Because it is required that information move across security boundaries, therefore by the Intermediate Value Theorem for Computer Security (CS-IVT), at least one multi-level component must exist [20, §6.2]. Distinct from firewalls—which they superficially resemble—and from Internet Protocol (IP) routers—which is how these devices appear to and integrate with the rest of the network—cross domain solutions might be said to operate above the Application Layer of the OSI network model, in the realm of security policy (see Figure 1.1).

[2]This is the same security model as used in Trusted Solaris 2.5.1 without the assertion of privilege. It was simplified in Trusted Solaris 8 and further simplified in Solaris 11 (with Trusted Extensions) by the elimination of information label functionality.

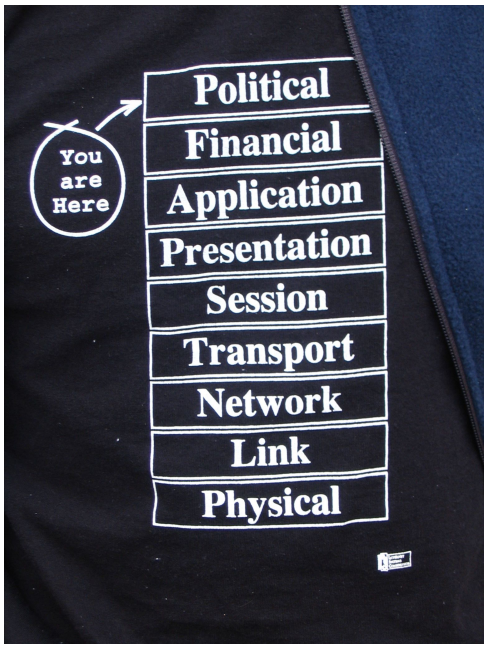


Figure 1.1: Nemeth's extension of the seven layer OSI network model (photo used by permission of Phil Wolff).

Cross domain systems are found everywhere, from military systems to medical records privacy to critical infrastructure industrial controls. Modern automobiles are cross domain systems. Although the fact is not widely appreciated yet, the chassis and under-hood information architecture, showing thought given to separation of safety-critical networks—that is, engine control, stability control, and braking—from non–safety-critical networks handling environmental controls and entertainment systems—is reminiscent of intelligence community cross domain security challenges and the same kind of attacks have been found to work there [50, 135, 257].

Cross domain systems occasionally embody international co-operation and communication in a box. If war is to be avoided, it is crucial that cross domain systems should work all the time and do exactly what they are

supposed to do, not less and not more. Consequently it is the job (1) of software and hardware developers to make the components of cross domain systems with high assurance; (2) of systems integrators to design systems and install components in the field with care and attention to security; (3) of certifiers and accreditors to test the security of components, systems, and installations thoroughly; and (4) of government programme offices—in the case of classified systems—to oversee the process.

It should not be thought, despite the focus of this thesis, that security evaluation of cross domain systems is only a problem of interest to military services or the intelligence community. The privacy of EHR needs be protected from excessive data mining by insurance companies for the purpose of unfair discrimination; eScience will be negatively affected if epidemiologists cannot compare histories because they remain locked up in isolated databases; and cars should not lose control and run off the road because of a malicious code transmission picked up by the tyre pressure monitoring system and routed to the off front brake actuator via an *ad hoc* cross domain system installed in the entertainment system by means of an authentication vulnerability in the firmware updating mechanism of the Engine Control Unit (ECU), just for example [50, §5], [135, §IV].

In factories, safety and *control* functions are commonly run on the same network, a departure from the recommended practice of only a few years ago: 'the biggest reason given... is to avoid common mode failures' [214]. But consolidation of at least two of three networks carrying Aircraft Control domain, Operator Information domain, and Passenger Entertainment domain traffic in several models of Boeing 737 aircraft has already begun; prior to Federal Aviation Administration (FAA) rule-making citing 'Special Conditions' in 2014, the three network domains previously ran on separate

busses, primarily for historical reasons[3] but that separation gave obvious safety benefits as well [75]. Since the adoption of DO-178/C, first the computers and now the networks have migrated to use of Virtual Machine (VM) separation on shared hardware, for equally obvious Space, Weight, and Power (SWaP) savings [89, 198, 200]. When building new airframes, aerospace engineers quite rightly wants to avoid running three wires where one will serve, saving kilograms of mass. But if we believe that VM separation suffices for safety-critical software, do we believe it as strongly for network traffic?

EHR systems in the U.S. in particular are in need of cross domain solutions. A few years ago, government funding initiatives encouraged most health care providers in the U.S. to adopt electronic record-keeping systems—with the ostensible purpose of reducing errors—and to convert their existing paper files to digital form. This has, for the most part, been done. But the U.S. at present lacks a central clearinghouse for EHR interchange; consequently, it is usually the case that different physicians' office's record-keeping systems cannot talk to one another, nor do they have direct connectivity to hospitals, nor to specialists. With the exception of a few Health Maintenance Organisation (HMO) systems like Kaiser Permanente, every time a patient is referred by a G.P. to a specialist, the patient's health care records go across on paper. It is obvious that tens of thousands of isolated EHR networks will be interconnected during the next few years; it ought to be done securely. The military solved this problem decades ago [156]. Information sharing between mutually distrustful data owners is isomorphic to the problem of protecting privacy whilst not precluding epidemiological studies beneficial to the health of the whole society, or the practice of health

[3]The three network domains came aboard years apart; it made retrofitting features like seat-back video screens and remote maintenance monitoring easier on old aircraft.

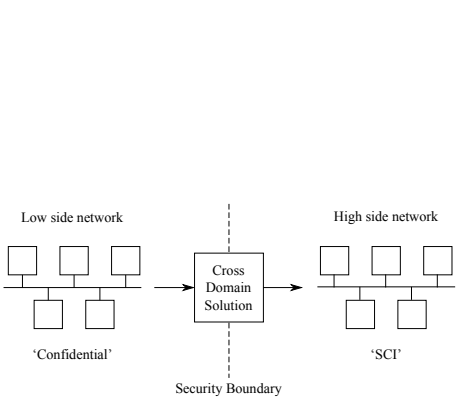
insurance.

1.2.1 Definitions

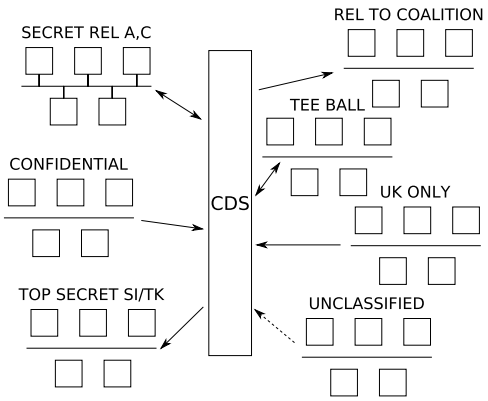
A cross domain system, abbreviated 'CDS', is the application of a multi-homed device interconnecting two or more security domains at different security classification levels. It is synonymous with 'Controlled Interface' [74]. A simple cross domain system, called a data diode, is shown in Figure 1.2(a). The one-way communication path is of limited usefulness, though, if only for the reason that it makes reliable communication across the interface impossible [207]. Real cross domain systems typically have a mixture of one-way and multi-way communication channels and multiple security classification levels, as shown in Figure 1.2(b).[4] The example in Figure 1.2(c) incorporates dual redundant cross domain solutions in a fail-over hot spare configuration with Virtual Local Area Network (VLAN) switches that have been evaluated for separation, and is typical of the level of complexity of cross domain systems being deployed today.

It is only recently, for about the last twenty years, that cross domain systems have been automated in the first place. In a sense, cross domain systems act like wartime censors, opening mail sent by soldiers at the front and excising a town name here, a code word there, or information about the readiness of forces. With the rise of electronic communication networks, cross domain systems evolved into an A/B switch between two teletypewriters, manned by a skilled operator whose job it was to read information from one network, throw the switch, and re-type it into the other network, occasionally with changes. This relatively primitive and extremely slow technology persisted until the early nineteen-nineties, when in the first Gulf War, to

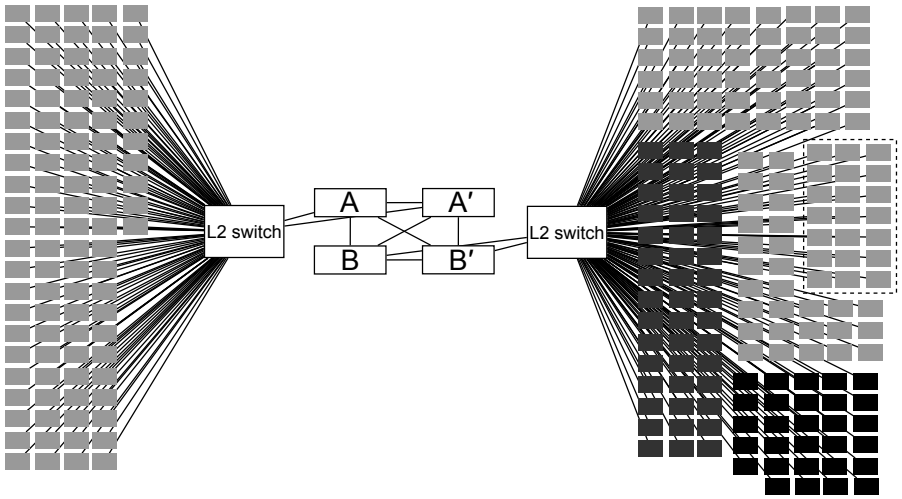
[4]Security classifications are for illustrative purposes only and do not necessarily indicate the presence of classified information.



(a) Simple cross domain system moving data only from 'low' to 'high'.



(b) Complex CDS exhibiting multi-level and multi-way data flows.



(c) Realistic example of a cross domain system taken from a real application.

Figure 1.2: Cross domain systems in conception and reality.

support a large international coalition, it was realised that a faster means of data interchange was required. Commanders in the field, early in the war, were forced by operational tempo to bypass administrative controls limiting connections between air-gapped networks, and some 'spills' naturally resulted. The first automated cross domain device, RADIANT MERCURY, was developed in response to that operational need. Because it was shockingly

unprecedented at the time to entrust the security release decision to an automated system, the certification testing of that first system was awesomely careful. Its lineal descendants are the subject of this narrative.

Radiant Mercury is a cross domain *solution*.[5] Unfortunately also abbreviated 'CDS', the term 'cross domain solution' as it is commonly used refers to a product, usually comprising certified software running on approved hardware, intended for the purpose of interconnecting two or more communicating endpoints at different security levels to form a cross domain system. The functionality of a cross domain solution may include one-way, two-way, or multi-way information flows, automatic sanitisation, upgrading, downgrading, and guarding. Other cross domain solutions may be simply data diodes [95, 96, 115, 141].

Cross domain solutions, uniquely combining inherent complexity with the potential to cause 'exceptionally grave damage to national security' should they fail, are developed and tested with the utmost care [180], [243, chapter 10, §5]. Beginning with formalised requirements and specifications review, proceeding through preliminary and critical design reviews, accompanied by high level and low level design documentation and configuration management of all hardware and computer software configuration items, systems engineering, and formal configuration control boards where records are kept, the software is written by vetted software development personnel in a closed area maintained under physical security [159]. Design and code inspections, automated static analysis of source code, unit and system level tests, documentation, training of software developers, installers, and operators, audits, and process improvement are all part of the expected

[5] To resolve the ambiguity, we suggest a compromise: retain the abbreviation 'CDS' for cross domain solutions and use 'XD' ('x-ray delta') to refer to what presently are called cross domain systems.

environment.

1.2.2 Two Analogies

To appreciate the difference between cross domain systems and cross domain solutions, it may be useful to consider the following analogy: the cross domain solution in a cross domain system is like the boiler in the infrastructure of a house. In Figure 1.3, corresponding to the unit isolation valves on the input and output of the boiler are the screening routers sometimes required by particularly conservative accreditors in classified environments, reflecting the accreditor's understanding of the principle of defence in depth through serial placement of devices with different implementations and functions. (More will be said about screening routers later.) The water inlet to the boiler, at ambient temperature, corresponds to the high side information system. The clean water supply corresponds to the higher value or higher classification security domain. The boiler acts as a cross domain solution, bridging the security boundary between the high side and low side with responsibility for protecting high side information from threats on the low side.[6] Protection devices on the boiler corresponding directly to security controls in a cross domain system include the Pressure Relief Valve (PRV), which fulfils the purpose of a commonplace security functional requirement on cross domain solutions, that the system must shut down automatically if audit storage becomes full. The flow meter corresponds to usage metrics, and a back-flow preventer, if fitted, to a data diode. The gas input to the burner corresponds

[6] As an aside, the heat input of the natural-gas–fired boiler can be considered, thermodynamically, to increase the entropy H (or, more properly, enthalpy, since the gauge pressure is approximately zero) of the low-side system, corresponding in Shannon's [1949] formulation to a reduction in the amount of information a receiver could understand from the signal, or more properly put, a smaller probability that the receiver's interpretation of the signal is correct, reflecting the operation of the sanitisation function of a cross domain system in the high-to-low direction [208].

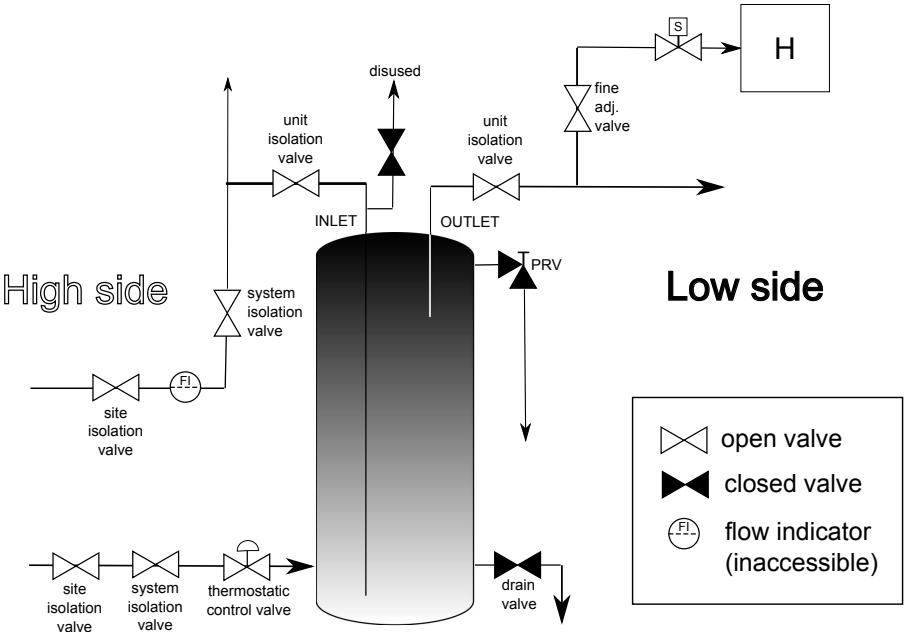


Figure 1.3: Piping and Instrumentation Diagram (P&ID) for the cross domain system analogy.

to sanitisation rules and the drain valve to the zeroisation function on the hardware that is required before allowing any hardware that has ever touched classified information to be shipped off-site for repair.

The analogy is imperfect but useful, because piping and instrumentation diagrams like Figure 1.3 are the starting point for another well-tested method of risk identification called HAZOP, used in the process chemical engineering industry [132]. By way of another analogy, HAZOP might be applied to the software development and certification & accreditation processes of a cross domain solution to locate potential 'hazards' posing a risk to the timely progression and budget of the software development and maintenance organisation and to the certification and accreditation of systems in the field. For example, following the HAZOP process, we first divide the unit to be

analysed into nodes. Like in chemical engineering, a node is a section of the process where a significant change takes place. In a chemical plant, a pump is a node because it increases pressure. In software development, commits in the software configuration management system (because they change the source code), design changes (because they change the product or process), board meetings (because they change the state of the project), and status reports (because they affect the project manager's internal model of the state of the project) are all nodes. As in HAZOP, we select a node, define its purpose, and determine the safe limits. Using guide words such as 'too much', 'too little', 'wrong material supplied', 'no flow', 'reverse flow', and others, we identify potential hazards and their causes. We determine how it is known that a safe limit has been exceeded. We estimate the consequences of each identified hazard, identify all existing safeguards, and estimate the frequency of occurrence of the hazard. Finally, we risk-rank the hazard. If done correctly, shortcomings in safeguards are identified ahead of an accident and disaster averted. Application of HAZOP to the software development and certification & accreditation processes is proposed for future work, but more will be said about the role of policy, process, and procedures and the relative maturity of software development and testing organisations and how it predicts the success or failure of a certification test activity after the development phase, in Chapter 6.

1.3 The Difficulty of Installation

Once released from the software development organisation and having completed certification testing, every cross domain solution must be installed in the field as part of a cross domain system. It is illuminating to consider the plight of the cross domain system installer as a problem in field philology.

Beginning back at the developer's plant, from the first contacts between the assigned installer and the first of several data owners to be affected by a particular cross domain system installation, and extending through accreditation and approval-to-operate, it is the installer's responsibility to understand the aggregation of all the different data owners' requirements for protection of their own information, not to mention the data formats for presentation and transliteration, if required. Data owners do not trust one another with access to their information, hence the need for a controlled interface between security enclaves in the first place [143].

It is not uncommon for the installer to have to negotiate specialised shipboard terminology one week, airborne or space-based systems on the following installation, complex pre-existing computer room setups where mutually distrustful data owners come together only reluctantly, or even war zones. An example may serve to set the scene. The following scenario is typical of installer guidance for the simplest and most straightforward of cross domain system installations, a 'type accreditation' of familiar hardware and software to be installed in a relatively standardised environment, comprising comparatively well-debugged communication interfaces and message formats:

> USS *Theodore Roosevelt* (CVN-71), a 104 000 tonne nuclear powered aircraft carrier of the *Nimitz* class, arrived in 2007 at Norfolk Naval Shipyard (NNSY) in Portsmouth, Virginia for a Planned Incremental Availability (PIA)—meaning a type of overhaul short of drydocking—of nine months duration. One of the scheduled activities during the 2007 PIA was installation of new computer systems in the carrier's Combat Direction Centre (CDC), located on the O3 deck to amidships forward of the island, adjacent to the Carrier Air Traffic Control Centre (CATCC) and

immediately below the ship's flight deck.

In the CDC spaces of a warship are found the processors of the AN/UAQ-119E(V) Global Command and Control System–Maritime (GCCS-M) battlespace management system[7], charged with maintaining portions of the ship's Common Operational Picture (COP) on the displays in the CDC. Aboard 'big decks' with their multifarious connections to communication networks at security classification levels ranging from Sensitive Compartmented Information (SCI) down to unclassified and foreign national sources, GCCS is protected from some of those networks—and other networks protected from GCCS—by a 'guard' or Controlled Interface (CI), conceptually a firewall, but more complicated than a packet filter and comprising one or more cross domain solutions.[8] During the 2007 PIA, the *Teddy Roosevelt* was to receive a CDS upgrade to markedly improve throughput of GCCS and to replace old, out-of- maintenance hardware and software with up-to-date versions. It did not happen that way.

'T.R.' waited in port as long as she could for the arrival of the specialist contractors who were to install her new CDS and of the Independent Verification and Validation (IV&V) specialists who were to test the system prior to granting of Approval to Connect (ATC) and Approval to Operate (ATO) by the Navy accreditor. The ship sailed at the end of the PIA without its GCCS-M COP fully operational because the cross domain solution was not

[7] GCCS is pronounced *geeks*.

[8] Security assurance and testing requirements for Top Secret/SCI and Below Interoperability (TSABI)-approved cross domain system interconnections in the U.S. defence department are established jointly by the National Security Agency (NSA) under the auspices of the Unified Cross Domain Management Office (UCDMO) and, in the case of afloat systems, by the Navy Cross Domain Management Office (NAVCDMO).

available in time. Months later, an expensive underway installation trip was mounted instead; the installer, trainer and IV&V team were flown to the ship by fixed-wing Carrier Onboard Delivery (COD) aircraft, then CODded off again at-sea afterwards.

Reasons for the delay of this representative CDS installation were complex, inter-related, and interesting. Factors included microprocessor vendor hardware product life-cycles,[9] operating system vendor software product life-cycles,[10] international information security evaluation and testing standards,[11] activities of U.S. and foreign companies,[12] minutiæ of government acquisition rules,[13] U.S. Navy and U.S. Department of Defence (DoD) policies,[14] and members of the intelligence community. In all, the *Roosevelt* waited almost seventeen months for her new cross domain system, most of that time taken up by Certification Test and Evaluation (CT&E) of a new version of the software, which had been forced to be developed outside the 'natural' software development life-cycle by events outside the scope of the developer's software engineering process.

The situation just described is not an anomaly [45]. It is unfortunately the routine consequence[15] of software engineering and CT&E processes currently

[9]The UltraSPARC III microprocessor with copper (Cu) interconnect was microcode-incompatible with the previous stepping of that chip that used an aluminium (Al) metalisation layer. This meant that Trusted Solaris 8 4/01 would not run on Cu machines.

[10]The certified version of Trusted Solaris 8 had no rating maintenance waiver for use with regular Solaris 8 HW 12/02 patches that enabled use on the UltraSPARC III Cu chip.

[11]The Common Criteria certificate for Trusted Solaris 8 (Certified Edition) specified a limited set of approved hardware in the Security Target, all of which were past End-of-Life (EOL) and no longer sold by the hardware vendor.

[12]The hardware manufacturing division of the OS vendor was located in California, but Trusted Solaris 8 was developed in the U.K., and certified by the Canadian government.

[13]NSTISSP No. 11 requires 'National Security' computer systems to have been evaluated and certified under the Common Criteria or equivalent.

[14]Recertification of the cross domain solution was triggered, under UCDMO rules, by the change of OS; as the certified hardware platform was unavailable for purchase new, it was briefly considered to acquire used hardware on the secondhand market, but this was rejected on the grounds that pedigree of the uncontrolled hardware could not be established.

[15]'. . . an acquisition process that Deputy Defense Secretary William Lynn told a Strategic

employed to develop and field critical systems in the defence departments of the U.S. and U.K. governments and as regards their relationships with certification and accreditation authorities who often are members, not of the military, but of the intelligence community [62, Chapter X].

1.4 Certification and Accreditation

Once the software and hardware of a cross domain solution have been developed, and have passed the developer's Factory Acceptance Test (FAT) procedures and have been shown to be functionally operational in satisfaction of the design requirements and specification, the hard work of security certification and accreditation begins. Before it will be deemed trustworthy by users, the product must be tested again for the purpose of certification, then installed in the field and each installed system tested again in the environment it which it will operate, in a process called accreditation. The reason for this is because the developer's own tests are designed to verify traceability from requirements through specification to design and functionality of the resulting system, but security is often a non-functional requirement [82, Chapter 2]. Certification and Accreditation (C&A)[16] in this context refers to a combination of:

Command Cyber Symposium audience takes "81 months from when an IT program is first funded to when it becomes operational." He added that "this means systems are being delivered four to five generations behind the state of the art." ' [114].

[16]The new preferred terminology, replacing C&A, is *Assessment and Authorisation (A&A)*.

- security testing of components, including:
  - regression testing to functional requirements, usually performed by the developer and witnessed by the certification authority;
  - independent verification and validation, performed by a different group from the developer; and
  - penetration testing, performed by the certifier;

  } 'Certification'

- re-testing of systems composed of certified components after the components have been installed in the field but before granting of approval for the system to connect and approval to operate; and

  } 'Accreditation'

- regular, required re-examination and re-authorisation of the entire system.

  } 'Maintenance'

Some certification schemes, particularly the Common Criteria, explicitly include requirements to validate that the developer's processes and procedures are adequate to provide a defined level of assurance. This is reflected in the distinction between Security Assurance Requirements (SAR) and Security Functional Requirements (SFR) throughout the Common Criteria [58]. Finally, as a result of lessons learnt in the certification of Compartmented Mode Workstation (CMW) products, virtually all security certification

schemes in use today, dating from the orange book, explicitly include a rating maintenance phase [242].

The certification phase of testing, formally called Certification Test and Evaluation (CT&E), precedes the installation of a cross domain solution in the field. CT&E is followed by Security Test and Evaluation (ST&E), or accreditation. As described in a previous paper:

> In the case of [cross domain solution] applications having field-alterable rules, CT&E is performed using a representative set of processing rules designed to exercise the capabilities of the system. In practice, the software developer (being the one most familiar with the system) creates an initial set of tests and expected results based on Factory Acceptance Test (FAT) procedures and test coverage analysis. Then a different organisation is engaged to use those procedures to perform Independent Verification and Validation (IV&V) and penetration testing on the system. At each stage in the certification process, *findings*—deviations from expected results—are reported to the developer and the certifier.
>
> After CT&E, each instance of the [cross domain solution] needs to be installed and accredited for a particular use in a particular location. After the initial site survey, trained installers configure rule sets in coordination with all of the data owners involved and set up the system in the location where it is to be used, though it is not connected to all of the network endpoints yet. At this time, site operations personnel, system administrators, and security officers are trained on the new system. Before the [cross domain system] is allowed to connect for the first time, it needs to be tested one more time in a process known as Security Test and Evaluation (ST&E) for accreditation. Since the ultimate purpose of a [cross domain solution] is to reduce residual risk to a level acceptable by the data owner(s), in each case there is a Designated Approving Authority (DAA)—a person delegated by the Principal Approving Authority (PAA) of the data owner formally to accept responsibility for residual risk in the operation of the [cross domain system]. IV&V contractors assist the DAA with ST&E by exercising the [cross domain solution] through test procedures specific to the site until the DAA is satisfied and agrees to accept responsibility for the residual risk. The DAA then issues an Approval to Connect (ATC) and allows the system to operate. The formal Approval to Operate (ATO) is for

> a limited time—no more than three years—and contingent on security-relevant changes not being made to the system without approval [142].

A cross domain solution product is *certified*, but every individual installation of that product must be independently *accredited*. What is certified is always a particular release of a particular software product running on specified hardware (the 'evaluated configuration'). Minor changes to certified software in a cross domain solution might be accommodated with extensive documentation; any kind of major change to the software, including new features or porting to a new operating system or hardware architecture, trigger immediate invalidation of the certificate and re-certification of the cross domain solution under all known certification schemes.

For 'cookie-cutter' installations,*i.e.*, those of identical hardware and software in identical network and physical security environments with equivalently cleared users, sometimes it is possible to secure a 'type accreditation', but those are uncommon. Type accreditations are suitable for shipboard installations across the same class of relatively small ship, *e.g.*, destroyers. They have not been found to be useful in ground stations or big decks, in general, because of differences across sites in the local network environment.

1.4.1 Moving Towards Reciprocity

At the 2011 Unified Cross Domain Management Office (UCDMO) conference, the following working definition of 'reciprocity' was given:

> If you bring me your documented evidence package, showing what you tested, how you tested it, and documentation that you did the testing the way you say you did, then I'll look at it and see what you've already done. I am not going to re-do tests that you have already done and documented that you did.
>
> Reciprocity does not mean accepting others' *authorisation*, only the documented evidence of tests already done. If the new

> agency's own risk assessment differs from yours and calls for additional tests, then only those additional tests will be run and must be run. [46, emphasis in original]

Reciprocity has been slow to arrive within the intelligence community and between the IC and the military services. The reasons are primarily historical, arising from differences of philosophy over protection of sources and methods of collection [192]. But the unreasonably high cost of certifying new cross domain solution products, and of accrediting new cross domain systems—both in money and time—are finally beginning to drive progress.

The new preferred terminology, originating in NIST SP 800-39, is 'Security Assessment and Authorisation', abbreviated A&A or SA&A [162]. 'Certification and accreditation', it was said, implied a finality that misled too many people into thinking they could simply:

> '... try to pick up an accredited system and put it down somewhere else. The new name correctly reflects a security authorisation process' [259].

The use of the new term also avoids confusion with the current abbreviation 'CNA' meaning Computer Network Attack [77]. There is talk at the annual UCDMO conference of fully integrating A&A into the acquisition process some day; security controls would then simply become contract requirements, written into the request for proposals and treated as deliverables in the same way as functional requirements [46].

1.4.2 The Particular Difficulty of Cross Domain C&A

The special difficulty of cross domain solution certification and cross domain system accreditation lies inherent in the fact that—by definition—such systems always will span at least one boundary between security domains controlled by different data owners [143]. Certification authorities, which

for historical reasons often originated in or eventually were placed under the bailiwick of intelligence-collection agencies, represent in their testing the concerns of their most conservative interests [2, 176]. Accreditation authorities vary in this style more than certification authorities, but certified cross domain solutions regularly encounter the trap concealed in that particular thicket as well, *i.e.*, the necessity to satisfy during accreditation the overlapping assurance criteria of multiple Designated Approving Authorities (DAAs), who represent the interests of data owners in the field. For the installer or developer to attempt to take an inclusive approach and test once to the 'high water mark' of the applicable security standards typically fails because of the lack of reciprocity. Consequently, the testing burden on the cross domain solution or system tends to be multiply duplicated as if the principle of defence in depth were conflated with the practice of Independent Verification and Validation (IV&V) [142]. 'With multiple data owners come multiple DAAs. With multiple DAAs come repeated rounds of ST&E, typically conducted by the same IV&V contractors—who, being already familiar with the system are the logical ones to test it at a reasonable cost —and using similar or identical test procedures' [*ibid.*]. That, simply, is the root cause of the unnecessarily high cost, in terms of both money and time, of cross domain certification and accreditation. The direct result of it is delays such as the one that left USS *Theodore Roosevelt* partly blind in wartime for almost a year and a half because of certification delays. The present situation is intolerable.

1.5 Research Problem

The research questions are summarised in Table 1.1.

Deployment of cross domain systems and solutions is expensive and

	Question	Location
RQ-1	Is C&A more expensive than it needs to be?	§6.2.2, §6.3, and §6.3.1
RQ-2	What can be learnt from a failed evaluation?	Chapter 4, §5.5.2, and §6.3
RQ-3	What can be learnt from a successful certification?	Chapter 5 and §6.2.7
RQ-4	What are the differences between the two that suggest a causal link from actions or inclination to action—or inaction—and outcomes?	§5.5.2, §6.3, §6.3.1, and §6.2.7

Table 1.1: Summary of research questions

time consuming, but it is more expensive and more time consuming than it needs to be. In point of fact, information sharing opportunities are foregone, underutilised, or missed in the flow of world events. We acknowledge that such systems combine functional complexity with a uniquely high risk of failure, and it is also true that the operational environment they are installed into presents challenges not facing commercial systems, but the method of testing and security evaluation presently employed is unnecessary duplicative of effort. It is no wonder cross domain systems are cumbersome to get accepted for approval to operate. But the cost is out of proportion to the actual level of information assurance achieved.

As a first step towards solving the problem, this thesis examines two closely related case studies. One is of a successful security certification of a new cross domain solution—actually it was a major software update of an old product but still an update that involved porting to a new operating system, a new hardware architecture, and significant new functionality at the same time. As a bonus, the certification criteria were unfamiliar to

both the developer and certifier as well; they had to familiarise themselves with a new standard as they went along. The other case study was an *un*successful Common Criteria security evaluation—here again, the developer had extensive experience with previous security certification criteria but none at all with the Common Criteria. That in itself is not an unfamiliar situation for cross domain solution developers and vendors to find themselves in; cross domain solution developers and installers regularly encounter operational situations in the field that have no clear precedent yet need to be solved rapidly, and furthermore it is the usual lot in life for a cross domain solution developer to encounter repeatedly new security testing criteria that are unfamiliar or which have suddenly changed [142]. As a former head of the company put it (in the context of hardware systems),

> But if the life expectancy of a regulation is so long as to inspire mortal policymakers to create them, it is much too *short* to provide the continuity needed to assure a stable lifetime for the average project. As noted previously, the average major item of defense-related hardware requires 8.3 years merely to develop, and remains in the inventory for 23 years thereafter in the case of an airplane and 33 years in the case of a ship (even excluding the lag between the end of development and the production of any individual item). Thus, viewing the development phase alone, there is in that period an average of one complete turnover of the regulations which were in being when a project was initiated, resulting in a whole superstructure of totally new regulations imposed subsequent to its birth. This is akin to changing the rules of a basketball game at halftime. Compound this regulatory turbulence during these formative years with the three aforementioned "streamlined" senior management decision-making council meetings, four successive sets of senior government officials, two sets of corporate leadership, eight budget cycles, and 144 votes in Congress on funding, and the miracle is not that many programs fail to survive but rather that some programs actually survive to fail [11, p. 334–5 (emphasis in original)].

Although the identity of the product and names of participants have

been anonymised, it is fair to reveal that it was the same product, although years and versions apart, that is the subject of both case studies. How could a system with nearly twenty years of successful pedigree, at the time of the failed Common Criteria validation—and which had withstood numerous and rigorous examinations at the hands of NSA among others at many times in the past—have possibly failed what ought to have been a relatively easy certification? The reasons why it should have been easy and why it was not make for an interesting story and will be explained in Chapter 4.

1.6 Thesis

The main points of the thesis are summarised in Table 1.2.

Thesis	**Key Idea**	**Locations**
1	Mutually distrustful data owners	§1.3, §1.9, §6.1.1, §6.3, and §7.1.1
2	Conflation of principle with practice	§1.1.2, §1.4.2, & Chapter 7
3	The view from different security clearances	§6.2
4	Grounded theory of communication channels in CT&E	§5.5–§5.5.4
	Model of accreditor behaviour (analytical)	§6.2–§6.2.6
5	Model of certifier behaviour (observational)	§6.2.7
6	Proof that accreditor channels exist	§6.2.3–§6.2.6
7	The security paradox	§6.2.4

Table 1.2: Summary of thesis points

1. It is an unappreciated fact that the difficulty of cross domain solution certification and cross domain system accreditation is inherent in the fact that—by definition—they uniquely always span at least one boundary between security domains controlled by mutually distrustful data owners, as justified in §1.3 and §1.9, analysed in §6.1.1 and §6.3,

and reiterated in §7.1.1. This makes them a suitable subject for study.

2. People involved in the certification—and especially the accreditation—processes seem to conflate the duplication of effort currently happening in the practice of cross domain solution (and system) IV&V with the security principle of defence in depth, which it is not. This is probably because the way certification and accreditation schemes are standardised presents the appearance of irresistible cost savings to managers. The resulting overexpediture of time and effort—caused by the previous point in the thesis, it will be argued in §1.4.2—yields no attendant increase in security assurance.

3. Even more duplication of effort arises from the structurally limited view of the cross domain system to be had by accreditors with different security clearances and responsibility for information at different classifications because the accreditors are constrained from communicating freely with other accreditors in order to agree upon the true level of residual risk. This assertion will be justified in §6.2–§6.2.2.

4. The key to solving this problem is to understand the explicit and implicit communication channels that exist amongst accreditors at different security levels, and between developer and certifier (including the professional security testers), through which people arrive at a shared comprehension of the level of residual risk in a cross domain solution certification or a cross domain system accreditation without violating the global security policy. Towards this end we present a grounded theory of developer–certifier interaction during the CT&E phase and an abstract model of inter-accreditor communication that is sufficiently general to reason about cross domain systems interconnecting security

classification levels that are both hierarchically and non-hierarchically related; *i.e.*, it is powerful enough to handle international accreditations. The grounded theory is described in §5.5. The accreditor model is analytical in nature, and is explained fully in §6.2.

5. Observationally, we derive a second model of certifier behaviour. It is much less powerful than the accreditor model but more immediately applicable; it offers the cross domain solution developer a means for exerting a measure of control over the schedule—and hence the cost—of certification testing, in a specific example of the more general observation that some evidence exists to suggest a new measure of organisational maturity for software development organisations that can predict the success or failure of security certification activities. The evidence for this is shown in §6.2.7.

6. In §6.2.3 to §6.2.6, the communication channels predicted by the accreditor model are shown to exist in the real world. Furthermore, they satisfy the criteria for 'reliable signals' established by Akerlof, Spence, and Stiglitz [1, 212, 223].

7. In the process of exhaustively checking every reachable configuration of the general accreditor model, a flaw in current official government security policy was found; it is conventional wisdom that, to avoid as far as possible the inevitable 'ratcheting up' of security classifications that happens in the Bell–LaPadula model, all personnel should be cleared to the lowest security level consistent with accomplishing their jobs. In the case of accreditors, however, this precaution can backfire in interesting ways; certain undesirable information flows are forced, and certain desirable information flows are inhibited. If, however, all accreditors

are simply cleared to the highest security level, no information leakage occurs. Paradoxically, it is sometimes the case that relaxing security rules improves security. (See §6.2.4.)

Thesis	Contribution	Location
1	The essential difficulty of cross domain C&A	§1.3, §1.9, §6.1.1, §6.3, and §7.1.1
2	The conflation of security practice with principle	§1.1.2 and §1.4.2
3–4, 6	Grounded theory and accreditor behaviour model	§6.2, §5.5–§5.5.4, and §6.2.3–§6.2.6
5	Certifier behaviour model with practical application	§6.2.7
7	Paradox in some government security recommendations	§6.2.4
—	The historically overlooked role of John Wilkins, the first certifier of IT systems in the 17th century	§2.2.1

Table 1.3: Summary of contributions

1.7 Contributions

This thesis makes the following contributions (Table 1.3). Firstly, it declares that one unique characteristic of cross domain systems is that they are invariably and by definition always installed in environments that are the collective responsibility of at least two mutually distrustful data owners. Secondly, it makes the case that it is this very environment that causes certifiers and accreditors to conflate the principle of defence in depth with the practice of independent verification and validation, leading to unnecessary duplication of work because of certain irresistible cost savings apparent to planners and managers. Thirdly, we propose a model of inter-accreditor communica-

tion based on the limited visibility that accreditors with different security clearances—not necessarily hierarchical clearances, it is important to point out—have of the two sides of an ideal cross domain system; the accreditor model is analytical in nature. Fourthly, we present a grounded theory of intra- and inter-group communication observed during the certification phase, and then show that the accreditor model is sufficiently general to reason about real cross domain systems; the accreditor model is powerful enough to handle international cross domain system accreditations. To complement the accreditor model, fifthly an observationally derived model of *certifier* behaviour is developed, derived from the grounded theory, less powerful than the accreditor model but more immediately applicable to the developer's needs. Sixthly, it is shown that the accreditor model predicts the existence of signals that have been observed in the real world, and furthermore these signals satisfy the criteria of Akerlof, Spence, and Stiglitz for reliable signals in the presence of asymmetric information. Seventhly, we use that observation to show that there exists a flaw in current official government security policy. Finally, some overdue credit is given to one historically ignored pioneer of information security certification in the seventeenth century.

1.8 Organisation

Chapter 2 reviews the literature of western government computer and communications security standards and their application to the certification and accreditation of cross domain solutions and cross domain systems, along with some literature outside the area of computer security that offer potential solutions to the high cost of certification and accreditation that prevail today. Chapter 3 describes the research methodology. Building on this groundwork and the commonality of related systems, subsequently chapters 4 and 5

introduce a pair of related case studies designated R' and R'' respectively, calling attention to different experiences that occurred in view of the common heritage of the underlying code and continuity of the software development organisation, sponsoring government departments, and certification authorities and professional security testers. Chapter 6 pulls together the results described already, interpreting them in light of a theoretical framework for understanding certification and accreditation problems abstractly. Finally, in Chapter 7 results are summarised and conclusions that can be drawn are presented. Following is a glossary of specialised terms and abbreviations; Appendix A which lists all of the codes used; a separable Appendix B containing the original data necessary to reproduce the methodology, ATLAS.ti hermeneutic units, and analytical memoranda; and a separable Appendix C that contains de-anonymisation codes necessary to link the identities of the systems and participants to the original data for traceability.

1.9 Summary

This chapter introduced the twin concepts of cross domain systems and cross domain solutions and offered some insight from experience on the peculiar problems suffered by the cross domain developer, installer, vendor, certifier, tester, accreditor, buyer, and user. Cross domain systems exist to violate security policies but must do so reliably and controllably; they show up in a surprising variety of places, from surgeries to automobiles to CIA headquarters. They are more than firewalls, which they resemble.

Because they are used in high-threat, high-risk environments, cross domain systems are developed and tested to the highest standards, comparable to space flight reliability requirements. They can be reasoned about, however, by analogy to piping and valves. In fact, some cross domain solutions used

in particularly high threat environments are mechanically as simple as a unidirectional strand of optical fibre; with no return path provided, it cannot be compromised. It also cannot be used for Transmission Control Protocol (TCP) connections, though it can support Universal Datagram Protocol (UDP).[17]

Cross domain systems float in a sea of untrust. They are only grudgingly accepted by data owners, tested to extremes by certifiers, re-tested by accreditors, and blamed when the mail doesn't go through. Is it any wonder that the cost of them is unreasonably expensive? Part of that cost can be attributed to a misunderstanding about the reason for effectiveness of independent verification and validation testing. Site accreditors representing the interests of data owners at different classification levels have good reasons not to talk too much, but this limits their ability to converge on an accurate view of the true level of residual risk. Out of a pair of case studies of related but different cross domain systems, we derive a theory that explains some aspects of the problem. The theory predicts the existence of signals from the presence of asymmetric information, and the signals are shown to be reliable according to a well-tested theory originated by Akerlof, Spence, and Stiglitz. It is paradoxical that sometimes, to improve security, it helps to relax the rules a bit. Cross domain certification and accreditation are unnecessarily expensive, but the cross domain solution developer is not as helpless to influence the cost as previously believed.

[17] Several vendors of data diodes, notably Tresys Technology, Inc., claim to be able to maintain TCP connections across a one-way fibre. They do this by means of TCP/IP proxies on both ends. While packet loss across a metre of optical fibre inside a sealed box is, arguably, improbable, the result is *not* an end-to-end transport layer session; in particular, RFC 793 flow control, slow-start, and congestion control mechanisms do not work [119].

Chapter 2

Literature Review

> *'You know you're in trouble when the Russians are adding safety features to your design.'*
>
> —Maciej Ceglowski, on the evolution of STS [47, note 5].

One of the earliest written guides to Information Technology (IT) security as we think of it today—*i.e.*, the cryptographic and physical protection of the confidentiality, availability, and integrity of information or communication channels [246]—appeared in the year 1641 with the publication of a book by the Rt Reverend John Wilkins (1614–1672) titled *Mercury, or the Secret and Swift Messenger*. Wilkins' book is unique—and remained so for almost four centuries—because of the way it devoted equal time to exposing techniques that failed to work as it did to recommending good practices. For example, immediately following a complete tutorial in cryptography, Wilkins presented a short course in cryptanalysis. After extolling the advantages of speedy telegraphic[1] communication over a distance, he talked about how to ensure accuracy in communication channels; in doing so, Wilkins anticipated

[1] This is 'telegraphic' in the old sense of semaphores and signal fires, not the electrical telegraph of the nineteenth century.

Shannon's information theory [207]. It is unclear how much Wilkins invented, but in 1641 he described a five-bit binary encoding—though he called them *fingers*—for the Latin alphabet[2], ternary and quaternary codes using frequency-shift keying [265, pp. 73–5]; substitution and transposition ciphers, use of cryptographic nulls [265, chapter VIII], the aforementioned use of redundancy over unreliable communication channels to improve reliability and conversely the minimisation of redundancy to gain efficiency through constrained channels [265, chapter XX], a method for solving polyalphabetic substitution ciphers hinting at Friedman's index of coincidence [92], steganography, telegraphy, and data compression [265]. Wilkins, in 1641, identified all three of the uses for coding in *abstract* channels—for secrecy, for reliability, and for compression.[3] Wilkins states in the introduction of the book that he collected notes from many sources, but his was certainly the first book to put forth the information straightforwardly in English [37].

2.1 Information Assurance in the 15th and 16th Centuries

Wilkins cites as his two primary sources the *Nuncius Inanimatus* of Bishop Francis Godwin (1562–1633), a book which was concerned mostly with telegraphy [103, 178] although it touched upon information security [232, p. 93] in the context of the risk of interception of long-distance communication; and the *Steganographia*, written in 1499 by Johannes Trithemius (1462–1516), a German abbot, and which circulated in encyphered manuscript form for a

[2] Francis Bacon published in 1623 another five-bit binary code method and it is reasonable to assume that Wilkins would have read the earlier book [110, pp. 79–80].

[3] There are other uses for codes in *instrumentalised* channels, primarily for the damping of side-band radiation, harmonics, inter-modulation, or DC-reference drift, *e.g.*, use of differential Manchester line coding instead of Non-Return-to-Zero–Level (NRZL); and especially for clock recovery.

hundred years before anyone knew for sure what the book was about [48]. The latter was finally typeset and printed in 1606 along with another book containing the decryption key (this was the *Clavis Generalis Triplex*) thereby proving that the first two books of *Steganographia* were in fact a textbook of cryptography [235]. Book III of *Steganographia*, however, remained undecoded for another four hundred years,[4] resulting in the book finding a place on the *Index Librorum Prohibitorum* because the Catholic Church believed it to be a book about magic [231, p. 221].[5]

Trithemius wrote a later book about cryptography in 1508, the *Polygraphiæ* [236] that caused him a bit less trouble with the church [166, p. 147n67]; *Steganographia* may in fact have been part of the *Polygraphiæ*; Ernst (1998) cites de Vigenère, who referred to the books in the same context:

> *. . . presque tout l'artifice de la* Steganographie *et* Polygraphie *de l'Abé Tritheme. . .* [81, p. 337n42].

Wilkins, for his part, never cited de Vigenère directly, but the polyalphabetic substitution ciphers described in *Mercury* are similar to those of *le chiffre indéchiffrable* and it is likely that Wilkins was familiar with the earlier work [72], [265, chapter VI].

Aside from the mechanics of cryptology and channel encoding that were first being worked out at the time, it should also be noted that the Borgias in fifteenth century Italy were among the first to develop a professional information security apparatus, but it was not until Sir Francis Walsingham's highly professional operation in 1583–7 that modern counterintelligence techniques—combining human intelligence, cryptanalysis, interception of

[4]The decryption key to Book III of *Steganographia* was discovered independently by Reeds and Ernst in the twentieth century [81, 189].

[5]I was unable to find *Steganographia* in the last edition of the *Index Librorum Prohibitorum* prior to its abolition in 1966 by Pope Paul VI, but it definitely appears in an earlier edition of the *Index* from 1758 [80, 190].

signals, and traffic analysis—were used effectively in the service of Queen Elizabeth I of England [44, 116, 124, 171].

2.2 The Seventeenth Century

It is likely that Wilkins, writing in the middle of the seventeenth century, would have known about the decoded meaning of the first two books of *Steganographia*, but the problem with those, as Wilkins recognised, was that earlier writers like Trithemius and Godwin ascribed too much to the unmeasurable influences of the spirit world. Wilkins—a member of the Royal Society and, according to his biographers, a man who liked to experiment [224]—could and did analyse claims such as the sympathetic effect of a pair of lodestone-influenced needles later separated by a distance, and argued convincingly, from the known properties of magnetism at the time, that while such a mechanism might work at a range of a few centimetres, through empty space or diamagnetic materials, there was no scientific reason to suppose it would span oceans when needed [266, preface].

2.2.1 The World's First IT Certifier

It might be argued that Wilkins is irrelevant to the problem of information security, or that we face different challenges today. But in fact Wilkins was the very first information technology security certifier. He pioneered the application of the new *scientific* standards of proof to testing whether information security systems worked as claimed. The problem, as Wilkins recognised, was that vendors of snake oil sold their products without any objective evidence of effectiveness:

> I have heard of a great Pretender to the knowledge of all ſecret Arts, confidently affirm, that he himſelf was able at that time,

> or any other, to ſhew me in a Glaſs what was done in any part of the World; what Ships were ſailing in the *Mediterranean*; who were walking in any Street of any City in *Spain*, or the like. And this he did aver with all the labour'd Expreſſions of a ſtrong Confidence [265, p. 81].

Wilkins alluded to Sir Francis Walsingham's campaign against Mary, Queen of Scots:

> ... ſuch a Diſcovery would be of excellent Uſe, eſpecially for ſome Occaſions that are incident to *Stateſmen* and *Soldiers*.
>
> That the Ignorance... hath often proved Fatal, not only to the Ruin of particular Perſons, but alſo of whole Armies and Kingdoms, may eaſily appear... [265, p. 5].

And he took a position in the full-disclosure debate...

> If it be feared that this Diſcourſe may unhappily advantage others in ſuch unlawful Courſes; 'tis conſiderable, that it does not only teach how to deceive, but conſequently alſo how to diſcover Deluſions [265, p. 90].

...but with appreciation for the liability risks that sometimes go along with information security research:

> ...the chiefe experiments are of ſuch nature, that they cannot be frequently practiſed, without juſt cauſe of ſuſpicion, when it iſ in the Magiſtrates power to prevent them [265, p. 7].

Until the publication of Vernam's stream cipher patent in the early twentieth century, Wilkins' research, through the books of later writers like Falconer who drew on him [83], was among the most important sources of cryptographic techniques for information security available in the open literature [227, 251]. Baudot's five-bit teletypewriter code may have been credited to Gauss [216], but we observe Wilkins using it two hundred years earlier. Let us therefore nominate John Wilkins as an honorary father of information assurance through his being the first certifier and accreditor.

2.3 Twentieth Century

> *'Menzies was an amateur at a time when his adversaries were professionals'* [98].

Interestingly, despite the demonstrated effectiveness of Walsingham's native counterintelligence organisation in the sixteenth century, Western governments—as late as the Second World War—failed to learn from his example. Instead, through at least the first decade of the twentieth century, they depended upon the service of wealthy amateur spies who were often personal friends of the head of state [6, 60, 124, 191, 192]. In this respect the British Secret Intelligence Service (SIS) in 1914 began operating in a professional manner long before the American Office of Strategic Services (OSS) did.[6] However, both SIS and OSS were largely counterintelligence and covert action agencies [84, 139]. By 1919, in Great Britain and the United States, it was clear that a parallel capability, focused on signals intelligence and the analysis and development of codes and cyphers for both domestic protection and foreign interception, was needed. This was the beginning of the modern information security and assurance agencies.

In Great Britain, the core capability came from the Government Code and Cypher School (GCCS), which merged with elements of the Special Operations Executive (SOE) that had been dissolved by Churchill after the war, to become Government Communications Headquarters (GCHQ) at the end of World War II [36, 123]. In the United States, the corresponding new group was the National Security Agency (NSA), formed secretly by President Truman in 1952, together with the National Bureau of Standards as their public face,[7] in the guise of the National Computer Security Centre (NCSC)

[6] OSS was started by William Donovan in 1942; it later became the Central Intelligence Agency (CIA) in 1945.

[7] Later the National Institute of Standards and Technology (NIST).

[14, 15, 16]. These agencies, among the earliest adopters of information technology systems, became the primary guardians and gatekeepers of IA in government and the setters of civilian, military, and intelligence community information security standards today.[8]

2.3.1 From Isolated Hosts to Computer Networks

One more piece will complete the puzzle picture of what the world finally looked like in the early 1980s when the first serious attempts to define standards for information technology security validation appeared. Prior to the nineteen-seventies, computers largely existed only as isolated systems [4]. At the time, a combination of physical security, personnel vetting and training, and the dearth of information available to relative outsiders constituted significant barriers to attack [136, 138]. Military systems, because they operated on different and separate networks, were somewhat less visible—hence less vulnerable—to attackers in the sense that we think of attackers today [32, 126, 127]. The threat back then was from insiders with authorised access. The Semi-Automatic Ground Environment (SAGE), an important early military system, was using telephone modems as early as the nineteen-fifties, but not across the Public Switched Telephone Network (PSTN), or at least not on accessible regions of it [172, 187]. With the advent of teleprocessing on the PSTN [66, 117, 183, 253] however, remote attacks by a small but growing number of poorly funded, curiosity-motivated individuals became possible; the prevalence of attacks began increasing exponentially around 1980 [220]. When systems began to be connected to the ARPANET

[8] Britain recognised the importance of computers in information security before the Americans did. The development of the Colossus Mark I by Tommy Flowers at Bletchley Park—besides preceding the invention of ENIAC and the Harvard Mark I by a few years—happened under the purview of GCCS at Station X, in contrast to the development of automatic digital computers in the U.S., which was funded by the Army for ballistics calculations [41, 43, 65, 146, 182, 196, 202].

starting in 1969, the problem became acute and with the introduction of hundreds of millions of internet protocol hosts beginning in approximately 1993, it exploded [51, 107, 221].

2.4 Emergence of Government Certification and Accreditation Standards

For a period of about thirty years between 1949 and 1980, the primary government information assurance problem in the U.S. and U.K. was counterintelligence [6, 188, 225, 240, 268]. The civilian information assurance problem, by contrast, was largely confined to insiders attempting to commit financial fraud [173, 199, 205, 215]. Two things changed after 1980: the aforementioned spread of computer networks,[9] and the escape of cryptography into the open environment [5, 78, 85, 124, 204]. The old drivers of insider fraud and counterintelligence are still there; they have simply been joined by identity theft targeted against individuals and Distributed Denial-of-Service (DDoS) against corporations and governments.

What came out of the changes during the twentieth century was a clear need for effective information technology security testing, at first to be provided in the form of validated product lists and certification and accreditation programmes for government systems.

The civilian world was and still is slow to take up the cause of testing and validating information security solutions by standard means. Outside government, organisations are stuck with *ad hoc* defences comprising firewalls, anti-virus, and intrusion detection systems whose performance is described by sales advertisements and validated—inversely—by press reports of data

[9] Becoming a problem because of the network effect identified by Metcalfe [38].

breaches. The development of information security standards was driven by government agencies, primarily the military and intelligence community of Five Eyes: the U.S., U.K., Canada, Australia, and New Zealand [192, chapter 13].

The first government development of computer and network security standards began in the mid nineteen-sixties when Willis Ware (1920–2013), then at NSA, took notice of the security problems introduced by computer time sharing [269, chapter 20]. Before 1967 there were few papers published on information technology security, with the exception of highly developed and widely used commercial telegraph codes [23, 216]. Computer security first appeared in the open literature at the Spring Joint Computer Conference held in Atlantic City, New Jersey, 18–20th April, 1967; there Ware had to present his paper in the biomedical applications track because there was no track for information security [255].

As far back as the first world war, Signal Corps members knew that telegraph and field telephone lines could be eavesdropped, but the problem was not written about in the open literature. In the semi-open (or 'grey') literature (sensitive but unclassified technical reports circulated amongst government agencies and defence contractors), the mathematical foundations of what was to become one of the most influential computer security standards ever published were being set down by Bell and LaPadula in the early nineteen-seventies [19, 256]. The 'simple security property' and '$*$-property' that Bell and LaPadula defined became the basis for the 'orange book' that defined the requirements for a 'compartmented mode workstation' [242, 137]. It was ideally suited for handling classified information according to either the 'collateral' or 'compartmented' methods.[10] A few years later, in recognition

[10]The term 'collateral' when applied to classified information refers to a purely hierarchical classification scheme with unclassified information requiring the least confidentiality

of the fact of computer networks, a follow-on volume to the orange book, called the 'red book' was published as well [156].

Bell's follow-up paper, 'Looking Back at the Bell–LaPadula Model', published in 2005, explains the rationale for several apparently (to a software developer writing code for Trusted Solaris) mystifying design decisions in the orange book, such as the 'compatibility' of ordered security levels in the file system, which seems opposite from the way things ought to work. According to Bell, it was decided by management fiat against the recommendation of the authors, who wanted compatibility to flow the other direction. The paper contains trenchant observations on the practical difficulty of certification and accreditation and the opportunity to place high-assurance gateways in the narrow lanes between almost-isolated networks [20].

Several computer manufacturers eventually built products to orange book and red book standards and got them certified, but they were never commercially successful in spite of manufacturers' attempts to interest banks and other commercial organisations in setting up their own internal security classification hierarchies [228, Appendix B: *see* the sample commercial `label_encodings` file]. Few such products exist today, among them the open source SE Linux and Trusted BSD, Trusted AIX (IBM), and Solaris 11 with Trusted Extensions (Oracle).

protection and Top Secret requiring the most protection. Strictly speaking, a person with a high security clearance is allowed to see any and all information of low security classification as well as highly classified information. This was considered to be a flaw by the intelligence community, who defined 'compartmented information' as a practicable alternative. Compartmented information is classified according to code words that have no hierarchical relationship. People need to be 'read in' separately to every compartment they require access to. Military services primarily use collateral classifications and the intelligence community prefers compartmented information.

2.5 Certification and Accreditation Today

Throughout the military and intelligence community, certification and accreditation have always been a distributed activity, with no one agency or set of controls dominant. The U.S. intelligence community, under the Director of Central Intelligence (DCI) (later the Director of National Intelligence [DNI]), through the Intelligence Community Chief Information Officer (IC CIO), until recently operated under a set of certification and accreditation criteria known as DCID 6/3, which specified requirements but not a process; notably it failed to specify implementation details [46, 74]. There was never any such thing as DCID 6/3 certification, only accreditation on a site-by-site basis. The U.S. military until recently operated two parallel certification and accreditation programmes, one called Secret and Below Interoperability (SABI) that was concerned with collateral information systems, and another called Top Secret and Below Interoperability (TSABI) that dealt with compartmented information. SABI and TSABI are in the process of being merged now but it is a difficult transition, as the requirements for each differ significantly, and not necessarily in the way one might expect.

Under DCID 6/3, the Designated Approving Authority (DAA), working through the DAA Representative ('DAA rep'), accepted responsibility on behalf of the data owner based on the results of Security Test and Evaluation (ST&E) under the auspices of the Principal Accrediting Authority (PAA), who was always the DCI for any system handling Sensitive Compartmented Information (SCI) or Special Intelligence (SI), which are the norm for intelligence–community-connected systems.

Other agencies—among them numbered the National Reconnaissance Office (NRO), the National Geospatial-Intelligence Agency (NGA)[11], Central

[11]The U.S. National Imagery and Mapping Agency (NIMA) became known as NGA in

Intelligence Agency (CIA), Federal Bureau of Investigation (FBI), Defence Intelligence Agency (DIA), the Defence Information Systems Agency (DISA), and each of the branches of the armed services (United States Army, Navy, Air Force, Marines, and Coast Guard)—operated their own independent certification and accreditation programmes for non–community-interconnected systems [163].

The NSA is the largest reservoir of information assurance expertise in the U.S., having been given that responsibility the same way that GCHQ was. Certifications are not issued by NSA; again, like the Communications-Electronics Security Group (CESG) within GCHQ, they generally provide a technical opinion, although the NSA does retain full responsibility for Type 1 and Suite B encryption in the U.S., and shares Common Criteria national scheme membership with NIST [164]. The U.S. Department of Defence (DoD) now oversees a unified process called the Defence and Intelligence Community Information Assurance Certification and Accreditation Programme (DIACAP) that takes precedence in any situation where information might cross over collateral boundaries—that is, between military services or security enclaves—and DoD effectively controls certification and accreditation of the classified networks (SIPRNET and JWICS) under SABI and TSABI rules [8, 9, 10].

At the time of the case study described in Chapter 5 and up until quite recently, a cross domain solution had to be multiply validated—and, where possible, certified—by every different security evaluation authority that shared responsibility for the data to be processed by any cross domain system incorporating the cross domain solution. For intelligence community and collateral cross domain systems in 2006, this worked out to a requirement

2003.

for DCID 6/3 at Protection Level 4 or 5, plus SABI and TSABI combined [13, 170]. Work is currently under way to simplify the process for cross domain solutions under Intelligence Community Directive (ICD) 503, which specifies a set of security controls derived from NIST Special Publication (SP) 800-53 according to the risk management framework of NIST SP 800-37. ICD 503 will finally bring U.S. government intelligence community and collateral certification and accreditation into commonality with the non-classified side of the U.S. federal government, which operates under Federal Information Security Management Act (FISMA) rules.

2.5.1 Non-Intelligence Community Unclassified C&A

For unclassified but sensitive U.S. government information,[12] NIST maintains the Federal Information Processing Standard (FIPS) 140-2 and the related validation standard for hashed message authentication codes, FIPS 180. FIPS 140-2 is important because it applies to cryptographic module protections for unclassified but sensitive information [160]. Products undergoing Common Criteria evaluation in the U.S. may have to show FIPS 140-2 validation for their cryptographic functions, depending on the protection profile used.

2.5.2 The Common Criteria

The first new international security certification standard to emerge in the new century was called the Common Criteria, or CC. The Common Criteria were jointly developed by the governments of the U.K., U.S., Canada, France, Germany, and the Netherlands [58]. The Common Criteria Recognition Agreement (CCRA) lists twenty-five countries divided into 'Certificate

[12]NSA, not NIST, maintains oversight and keying responsibility of cryptographic modules for classified information through the Type 1 and Suite B cryptographic module programmes.

Authorising'[13] and 'Certificate Consuming'[14] members.

The certificate authorising members are the ones that matter from this perspective. Certificate-consuming members of the CCRA recognise each other's certificates up to and including Evaluation Assurance Level 4 (EAL4) universally, up to EAL5 within the NATO countries, and in the case of France and Germany between them, EAL7. Within the U.S., NSA prohibits use of validated products above EAL4 that were not evaluated in the United States by NIAP CCEVS [213].

As might be inferred from the preceding paragraph, the Common Criteria have been criticised for a vulnerability to being gamed, inconsistent stringency of different national validation schemes, the monopoly power of authorised testing laboratories, and several high-EAL certifications of products with little functionality or extremely tailored security targets [26, 96, 112, 113, 118, 121, 125, 158, 181, 209, 258]. Murdoch, *et al.* point to the same issues in the context of unclassified information in the payments card industry by referring to the fact that Common Criteria evaluation is a *design certification* in the terminology of the Ware report [153, 256].

[13]Certificate authorising members include Australia and New Zealand (the Australasian Information Security Evaluation Programme), Canada (the Canadian Common Criteria Evaluation and Certification Scheme), France (the *Direction Centrale de la Securite des Systemes d'Information* (DCSSI)), Germany (the *Bundesamt für Sicherheit in der Informationstechnik*), Japan (the Japan Information Technology Security Evaluation and Certification Scheme), Korea (their IT Security Evaluation and Certification Scheme), the Netherlands (*TNO Certification BV* [Netherlands National Communications Security Agency]), Norway (the *Sertifiseringsmyndigheten for IT-sikkerhet i produkter og systemer* [Norwegian Certification Authority for IT Security (SERTIT), which is operated by the NSM (*Nasjonal Sikkerhetsmyndighet* [Norwegian National Security Authority])]), Spain (*Organismo de Certificación de la Seguridad de las Tecnologías de la Información*), Sweden (the Swedish Common Criteria Evaluation and Certification Scheme), the UK (IT Security Evaluation and Certification Scheme, part of GCHQ), and the US (the Common Criteria Evaluation and Validation Scheme (CCEVS), run by NSA and NIST jointly under the National Information Assurance Partnership (NIAP).

[14]The certificate-consuming member countries comprise Austria, the Czech Republic, Denmark (*IT- og Telestyrelsen* [National IT and Telecom Agency]), Finland, Greece, Hungary, India, Israel, Italy (the *Organismo di Certificazione della Sicurezza Informatica* [Organization for IT Security Certification], Malaysia (called CyberSecurity Malaysia), Singapore, and Turkey (the *Türk Standardlarý Enstitüsü* [Turkish Standards Institution]).

2.6 Specialised Training of Personnel

The literature of certification and accreditation is wide but shallow. Accreditation, for government employees, is taught on the job and learnt from reading standards documents and reports. Insight into some of the challenges of working *in* the government may be found in memoirs from government laboratories and seem primarily to have to do with excessively rigid project boundaries and scientific communication [52, 62, 261]; conversely, in commercial companies employed on government contracts working *with* the government, it is government secrecy rules, auditors, and funding taps that seem to pose the greatest frustrations [11, 73, 146, 229]. Scientists in government labs have unusual freedom to talk to all of the contractors working in a particular field, facilitating communication that would otherwise not occur [Clark, Dequasie, Lundstrom *op. cit.*].

Certification standards documents are plentiful but mostly the available copies are obsolescent versions. There are no textbooks and few applicable certifications. One of the few certification and accreditation-relevant certifications available today is the Information Systems Security Engineering Professional (ISSEP) from the International Information Systems Security Certification Consortium [$(ISC)^2$]. ISSEP is a bit of a misnomer; the Common Body of Knowledge (CBK) of the ISSEP certification appears to be at least one-third Systems Engineering (SE) principles, and one-third Information Security Systems Engineering (ISSE), with the balance being U.S. government certification and accreditation public laws and standards applicable to a mixture of classified and unclassified environments. The ISSE curriculum is defined by the U.S. National Security Agency (NSA) in concert with its Centres of Excellence university sponsorship programme. Information security certification and accreditation standards from outside

the U.S., with the partial exception of the Common Criteria, are not taught. CISSP-ISSEP is, however, the only certification of its kind.

ISSEPs are well grounded in systems engineering and U.S. government certification and accreditation requirements, not only those used by the military and intelligence community, but civilian and commercial standards as well. The examination covers the latest incarnation of public laws and C&A programmes only; older versions are not included. This can be an obstacle to self-study; in 2009 the examination was updated to cover DIACAP while published study materials only taught the older Defence Information Technology Security Certification and Accreditation Programme (DITSCAP). Although strongly U.S.-biased, the examination is wide ranging, covering the gamut from Federal Information Processing Standard (FIPS) 140-2, to the Privacy Act of 1974, the Computer Security Act of 1987 (as amended), the Health Insurance Portability and Accountability Act (HIPAA), the NIST Special Publications 800-37 and 800-53, Office of Management and Budget (OMB) Circular A-130, and commercial Payment Card International (PCI) requirements. As of this writing, there are 788 ISSEPs in the U.S., two in the U.K., and fifteen in the rest of the world [120].

These certifications have three uses. For individuals, they can get help to get a person hired. For department managers, they are countable tokens useful for demonstrating to upper management that due diligence in hiring and training with respect to security has been done. For defence contractors, they are a necessity for bid and proposal work.

2.6.1 CDS Software Developer Certification

Related to the subject of certification of software development personnel on cross domain systems, U.S. Department of Defence directive DoD 8570.01

requires all personnel having privileged access to DOD systems or any software development responsibilities to have been certified in information security by the end of calendar year 2011. More than 100 000 people were directly affected. Compliance varies from 25 to 40 percent amongst defence contractors and parts of DOD, with a few military units reported as being fully compliant. The deadline has been extended several times but no more extensions are being given.

With the full effectiveness of 8570.01, in principle almost every CDS project could further require at least one person certified at the higher ISSEP (or ISSAP, a related certification for system architects) level to satisfy Information Assurance Workforce System Architect and Engineer (IASAE) requirements. The number of these presently in existence is relatively small. It is clear from Chapter 10 of DOD 8570.01-M (as of Change 3) that this applies directly to cross domain systems: in Table C10.T7, line IASAE-III.7 mentions CDS; lines IASAE-III.15–16 mention 'multi-level' and SCI. If the Level of Concern (LoC) for availability or integrity is high, or data crosses the Computing Enclave (CE) or Network Enclave (NE) boundary, then at least one IASAE Level III is needed on the project. The presence of Protection Level 1 or 2 indicates the need for an IASAE III; PL-3, 4, or 5 requires IASAE Level III.

In conclusion, it is clear that CDS software and hardware developers will have to be certified at the IASAE Level II or III, and not the lower Information Assurance Technical Workforce (IAT) level, because the description of the software development environment and the software product in 8570.01-M both precisely match the characteristics of a cross domain system.

2.7 Further Afield

Chemical process plant engineering, a field similarly at risk of dire consequences from accidents, has developed methods for analysing processes and plants in a systematic way to find opportunities where human error could cause accidents; it is a smaller part of the literature of safety-critical systems, but more closely applicable to software engineering than might at first be apparent. Economics and its application to risk assessment and risk management are another field where we have found useful techniques, particularly the job market signalling of Stigler, Spence, and Stiglitz [222, 212, 223]. There is an extensive literature on the subject of engineering failure, particularly software engineering failures [40, 67, 87, 100, 101, 102, 194, 195].

2.7.1 Chemical Engineering

Trevor Kletz has written extensively on the causes of human error and why organisations fail to learn the lessons taught by accidents [128, 129, 130, 131, 132, 133]. One of the distinctions Kletz draws in his books is between rule-based behaviour and knowledge-based behaviour; on the same topic, as Backhouse and Dhillon put it:

> Checklists inevitably draw concern onto the detail of procedure without addressing the key task of understanding what the substantive questions are. Procedures are constantly changing and for this reason offer little in the way of analytical stability [12, p. 2].

Procedures induce rule-based behaviour; rule-based behaviour has the advantage of rapid response to transient events but it is inflexible. Training induces knowledge-based behaviour; knowledge-based behaviour is able to adapt to circumstances [129, pp. 52–53]. Baskerville (1992) identified the same problem, in the context of information systems security, even earlier: he

noted that security practitioners (among whom, significantly, he numbered software developers and users in addition to 'system security officers' and testers) fell into—at the time—three generations of methods [18]. The three generations, which are believed by later readers of this seminal paper to be not entirely hierarchical, but overlapping, begin with 'checklist' methods, moving into 'mechanistic engineering methods', towards 'logical engineering methods'.[15] The individuals and organisations studied for this thesis are in all of the first three generations; penetration testers are clearly in the first generation, software developers and security certifiers in the second generation, and some of the newest standards such as NIST SP 800-39 are beginning to advocate for third generation methods. More will be said about this in Chapter 5.

2.7.2 Safety-Critical Systems

> *Safety has been at this a long time. It was a precursor to cyber security* [214].

There is disagreement among software engineering researchers whether safety-critical principles can validly be applied to security-critical systems [122, 133, 226]. One area in which safety-critical principles and software development overlap is spacecraft systems; Harland and Lorenz observe that systems designed with failure tolerance in mind—*i.e.*, to be able to work in degraded mode whilst in a non-deployed or partially deployed state—have saved many a mission [109]. But for a while, beginning in 1992, NASA went in a different direction; under the 'Faster, Better, Cheaper' (FBC) banner, NASA administrator Dan Goldin said,

[15] Baskerville hinted, in 1992, of a fourth generation, since reached, of 'usable security' methods; see for example [88].

> . . . that if failures do not occur, then NASA was not trying hard enough. . . [*ibid.*].

At least four major failures of robotic space missions occurred during that time: *Mars Observer* was lost in 1993; *Mars Climate Orbiter* and *Mars Polar Lander* were both lost in 1999, the Genesis lander in 2004 malfunctioned due to a requirements and specification error, and problems were still occurring in other NASA programmes as late as the end of that year [168, 154, 97]. Even in the face of a record so poor during those years,

> A more sanguine long-term view might have permitted a few more high-risk attempts, in order to achieve a better determination of the real reliability of lower-cost space systems. . . [Harland, *op cit.*].

FBC, which encouraged increased risk taking by projects to reduce cost, was abandoned, and the space agency's record of success has since improved [97, Appendix E].

Hard real-time requirements generally overlap with safety-critical systems, and modern processor architectures, because of the effect of data-dependent run-time cache behaviour, speculative execution, multi-threading and interprocessor communication (but not so much pipelining) make it very difficult to establish efficient Worst Case Execution Time (WCET) bounds on real-time software [111, Chapter 6], [175]. Cycle counting is ineffective; as Wilhelm and Grund put it, evocatively,

> The intrepid reader may think about the ramifications of "undead code," that is, unreachable code that gets executed speculatively; or, similarly, loop bounds that get exceeded by speculative execution [264, p. 99].

Such a prospect has negative implications for security as well as safety. Security test and evaluation, therefore, are even more important than generally

believed.

2.8 Summary

The development of the information assurance security process can be traced back as far as the fifteenth century. John Wilkins is clearly the ancestor of today's certifiers and accreditors of cross domain systems. Modern government, military, intelligence community, and some civilian security certification standards emerged from the seminal work of Bell and LaPadula, but credibility and applicability problems persist with all current security certification standards. Looking outside the field of computer security, some hard-won principles from safety critical engineering and space systems are relevant to the problem.

Chapter 3

Methodology

> *'We should always try to measure the property we wish to know directly, rather than measuring another property from which the property we wish to know can be inferred'* [133, p. 89].

Historically, this project originated with an engineering failure. From 2004–6, the U.S.-based developer D of a successful cross domain solution software product, designated here as R, worked to obtain a Common Criteria certificate for a specific version of the R software, called R', to be integrated into a larger system S that was being built for an overseas customer C.[1] Certain technology export control laws in the U.S.—the International Traffic in Armaments Regulation (ITAR)—had recently been relaxed to some degree, and D saw an opportunity to sell its product outside the country, a market that up to that time had been proscribed because R was on the 'munitions list' of un-exportable items. While unclassified, R was 'sensitive' and ineligible for export. R' would therefore be a limited functionality version of R intended for international sales (Figure 3.1).

[1] Appendix C associates code names used throughout with the actual organisation names, classified projects, and individuals involved. These codes may be used along with the anonymised data in Appendix B to recover the original data, if needed.

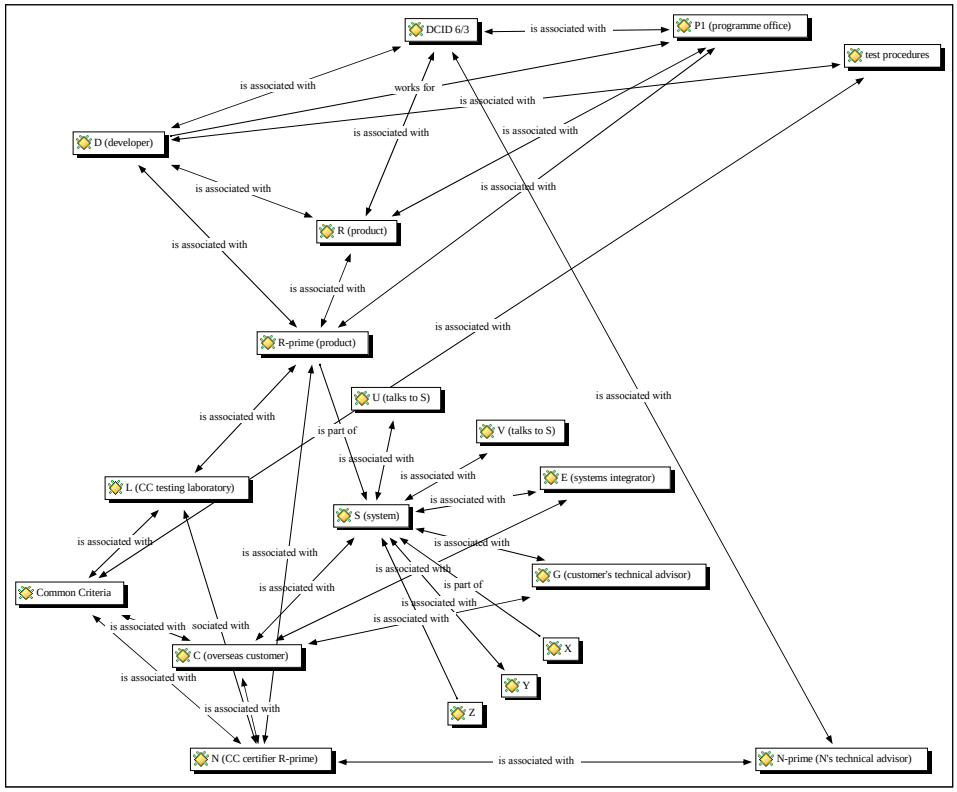


Figure 3.1: Overview of relationships between anonymised code names in the R' case study, chapter 4.

In the figure, R' is the software version under study, a version of R. It was developed by D at the request of a customer C to be part of a larger system S designed for C by system integrator E and composed of subsystem components X, Y, and Z. S, when completed, was planned to talk to other systems U and V. The customer's technical adviser was G. Testing laboratory L was engaged by the R programme office P_1 to perform the validation—using, as much as possible, D's existing test procedures—for certifier N; the developer had previous experience testing to the requirements of DCID 6/3, an earlier set of certification criteria, under certifier N'.

Figure 3.2 on the following page shows an overview of the names in the second case study, some of which are shared. The developer, D, along with the R government programme office P_1 and previous certifier, N', are all the same. This time, though, the product is R'', and the certification criteria have evolved to include the NIST security controls specified by DIACAP. A new IV&V contractor (P_2) and certification authority (Q), also make an appearance. Unlike events in the first case study, the second security certification of R was successful. Figure 3.3 on page 61 shows the relationship between names in both case studies.

The Common Criteria security evaluation of R' was expected to take about a year. The evaluation process began in the usual fashion with validation of evidence provided by D and guidance through the validation and evaluation project by L, a licenced Common Criteria Testing Laboratory (CCTL), but it was unsuccessful. To learn from that failure, and from the serendipitous availability of a contrasting success that nevertheless shared many of the same underlying elements, this research was initiated.

3.1 Attack

The purpose of this chapter is to describe the methodology of data collection, analysis, interpretation, theory-building, and validation used to make sense of the failure of the R' security evaluation in light of the success of R''. The methodology is tailored to answer the research questions in Table 1.1 on page 23 at the end of Chapter 1 and expanded here:

- **RQ-1** Are certification and accreditation more expensive than they need to be?
- **RQ-2** What can be learnt from a failed Common Criteria evaluation?

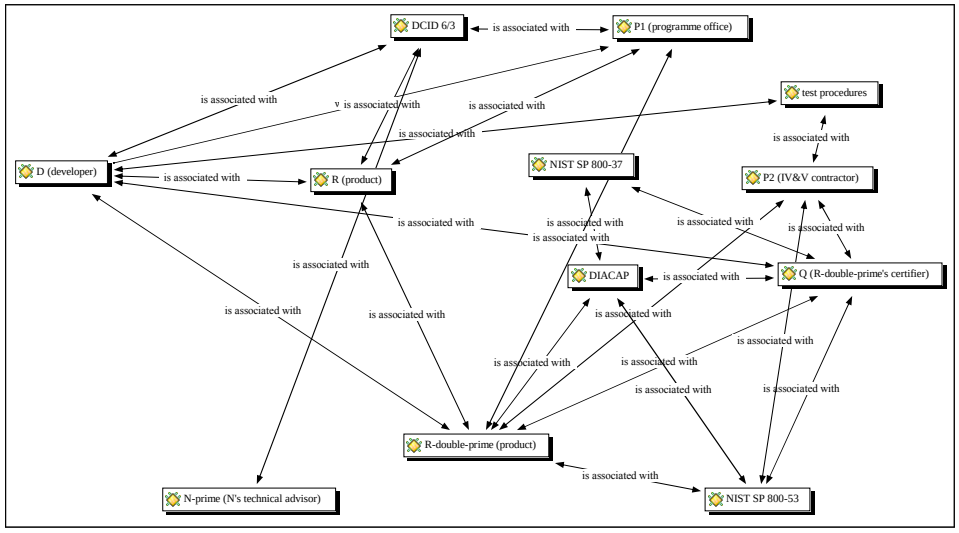


Figure 3.2: Overview of relationships between anonymised code names in the R'' case study, chapter 5.

- **RQ-3** What can be learnt from a successful DIACAP certification?
- **RQ-4** What are the differences between the two that suggest a causal link between actions or inclination to action—or inaction—and outcomes?

In addition it may be useful to go beyond the research data to try to discover new methods in answer to the following questions:

- **RQ-5** Is it possible that C&A can be made more effective?
- **RQ-6** Is it possible that C&A can be made more efficient?

where, in this context, *more effective* means that cross domain systems are tested better than they are now,[2] and *more efficient* means that there

[2]If the reader does not mind a forward reference, here is a hint of what 'more effective' will be shown to imply in Chapter 6: not only that undesirable information flows are blocked, as in the conventional definition of security, but further that desirable information flows are not inhibited.

are hidden opportunities to reduce cost. (It is recognised that the latter two goals may not be entirely compatible.)

3.2 Disclosure of Potential Conflict of Interest

The researcher was a participant in each of the projects described in the following sections, an employee of the corporation that developed the R' and R'' software, and a computer programmer on the R programme at D. The researcher worked on both projects, attended meetings in an official capacity, and was responsible for some code and documentation. Hereinafter when the 'thoughts' of D, being a department of a corporation, are described, they should be interpreted as the recollection of a participant in these events who was a member of D's software development and project and programme management team, a group of approximately twenty people. When meetings are described, either the researcher was in attendance at the meeting or heard or read a report of the meeting from another participant who was there. The researcher, however, participated at the level of 'staff software engineer', a technical position with no management responsibilities, and was not invited to every meeting. In the chronologically last to occur case study, however, the researcher filled a different role. While still an employee of the corporation and retaining the same access to the physical facilities and space and information of D, the researcher was working on a different project at the time and participated locally and remotely—as an observer only—in meetings and telephone conferences related to the certification of R'', a follow-on version of the R software. In return for access to the compartmentalised R'' meetings while not an active software developer on the project, the researcher took notes during meetings and gave copies of the transcribed notes to the meeting organiser. The relationship was mutually

beneficial, as the researcher collected data for this thesis at the same time the meeting organiser obtained detailed minutes of every meeting for free.

3.3 Anonymisation

Design and development of *R* were declassified in 1997 with the exception of still-classified relationships between *R* and certain agencies, but the programme remains physically located in a DoD closed area and all personnel are cleared to at least the SECRET collateral level. Because some *R* configurations are SCI, the closed area contains several additional Sensitive Compartmented Information Facility (SCIF) areas and some proportion of the software development, training and documentation, installation, test, and management personnel are cleared for SCI. To protect classified information, some relationships between *D* and *R* and certain U.S. government agencies have been obscured. To protect the privacy of individuals while permitting certain professional relationships to be discerned, all personal names have been systematically anonymised.

To protect proprietary information and competition sensitive information, names of companies, systems, and products, with the exception of Radiant Mercury, *Doxygen*, and UNIX, are anonymised throughout. Deanonymisation codes for all participants are supplied in a separable Appendix C.

3.4 Data Collection

Due to the limitations placed on the researcher by the fact that participants, events, and subjects existed inside closed areas (frequently SCIFs) and mixed with, discussed, or reported classified information, it was necessary to stray

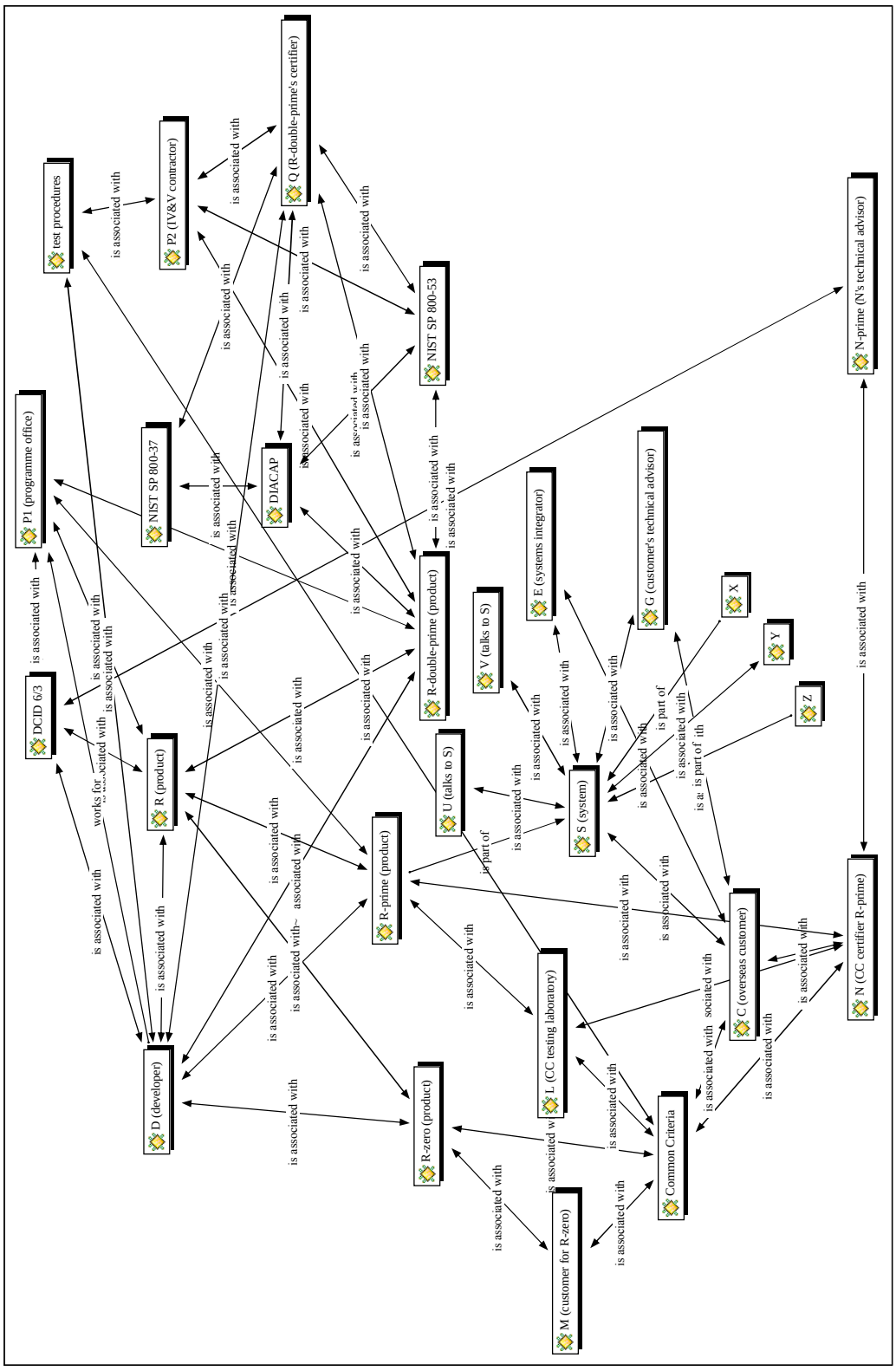


Figure 3.3: Overview of relationships between anonymised code names in all case studies.

from standard ethnomethodological practice. Direct quotations are not employed because of a blanket ban on recording devices at the site. (In the second case study, from which detailed meeting minutes were available, paraphrased quotations are used.) Reports, in general—including those written by the researcher in the normal course of duties—were almost always classified. Telephone conference calls attended by the researcher invariably took place over secure Voice-over-IP (VoIP) lines so that participants could freely discuss classified security vulnerabilities, risk assessments, mitigation strategies, and testing methods. Practically all of the evidence is the researcher's own notes and recollections of meetings participated in, documents and events seen, and personal conversations.

The situation is not entirely dissimilar to that described by Dalton as far back as 1959 about which he noted the dangers [70] of his methodology:

> The inquirer who labors from the screen of his formal nonresearch role may forget important methodological facts as he strives to enrich is data. He may learn much of unofficial activities and get behind protective screens but forget that official roles have official images. That is, in his speculative prowling he is almost certain at times to forget that nonintimates see his formal function as embracing only a limited knowledge of unofficial events. Eager to learn more, he alarms some persons, even his fringe intimates, by accidentally disclosing bits of unofficial information they think it strange that he should have. [71, p. 88]

The majority of the notes collected in the course of this research have the character of Glaser and Strauss's 'analytical memoranda' (section 3.6) combining the researcher's interpretation with observation of facts oftentimes obliquely referred to because of security classification rules. To avoid proprietary information difficulties, direct reference is never made to actual reports, budgets, schedules, plans, test plans, test procedures, emails, or presentations; only the researcher's notes, recollection, and reconstructions. As distinct

from the approaches of Whyte and Dalton to their respective populations of executive managers and industrial labour managers, the approach here owes much to that recommended by Mills in 1946 when studying engineers:

> These men are driven by the instinct of workmanship and by an emulative desire for recognition by their peers. They are rarely aggressive and never acqusitive. They like to work on a basis of cooperative equality and are not of the type to become executives, even minor ones... [w]ith such characteristics they are particularly subject to exploitation. [151]

The population of this study, comprising primarily software engineers with security clearances—even the programme manager, and all of the project leads, were educated in either engineering or physics before being promoted to management—is especially sensitive to propriety and rules of security, echoing the concerns of Dalton (*op cit.*) about the danger to the researcher of losing privileged access through over-extension. For this reason, the researcher maintained a passive role throughout the R'' project regarding the gathering of information and conducted almost no formal interviews, with the exception of participants T, K_1, K_3, and ivv_2 who were interviewed to validate the accreditor behaviour model, the certifier behaviour model, the effect that changes might have on the budget and funding of future projects, and validation of the models, respectively (Chapter 6). Secondary data from the project that the researcher obtained but does not refer to directly includes unsolicited email from two project participants throwing light on some events in R' that the researcher was not privy to, all of the reports written by the researcher during the R_0 and R' projects (most of which are classified or proprietary and cannot be reproduced here), but few intra-management documents such as budgets, detailed Gantt chart schedules, Task Order (T.O.) or Statement-of-Work (SOW) artefacts, because those documents are

not visible to staff software engineers in the business process used by D. After anonymisation, the data thus constituted are collected in Appendix B.

3.5 No Classified

The researcher had an active security clearance at the time of these events, has an inactive clearance now, and is subject to lifetime pre-publication review under title 18, United States Code, and title 50, U.S.C. [169]. This thesis contains no classified information; classification markings appearing herein are for demonstration purposes only, and do not indicate the presence of classified information. All of the information in this thesis and the researcher's previous papers were reviewed before publication by the appropriate government security officer.

3.5.1 Export Control

A letter was requested and received from the Export Control Compliance office of D laying out the identification and content of all D-proprietary information collected, retained, or copied by the researcher in the course of research—in many cases these were reports, white papers, or emails written by the researcher in the normal course of duties for D—comprising the data of case studies R' and R'', or the earlier R_0 project. All of these documents were kept in a pair of locked 35ℓ containers under the researcher's control. Records not in hard-copy form were kept encrypted on an external hard disk or DVD-ROM and stored in the same containers. Data required to understand and justify the conclusions of this thesis not subject to Export Control are reproduced in Appendix B.

3.6 Grounded Theory Methodology

Traditionally, to study the behaviour and interactions of a closed group of individuals in the field, researchers choose to do ethnomethodology [94]. In ethnomethodological studies, audio and video recordings may be used to observe very closely the behaviour of subjects, their speech patterns, use of language, and interaction with other people, equipment, and their environment; recordings are then reviewed over and over again, noting very small details. Ethnomethodology looks at what people actually do, not 'what they say they do' [145]. People may not be able to articulate clearly the underlying strategy of their actions[3]; nevertheless, an ethnomethodologist can discover it. Merely describing the behaviour of the studied population is insufficient, however; '. . . [one] rule applies to both qualitative and quantitative research—theory is more important than data. Theory is the contribution to knowledge' [69, p. 174]. Journal editors look for the existence of a theory in qualitative research paper submissions because '[theory] provides the story that gives data meaning' and '[theory] need not be formal or complex—theory should simply explain why' [69, p. 166]. The goal of the theory in the following chapters is to try to understand why the first project failed but the second one did not—putting forth a reasonable explanation for observations in light of the different experience, resources, knowledge, and goals of the standard roles involved in any cross domain C&A.

An example of a successful modern ethnomethodological-type study is the report on the London Ambulance Service, which similarly to this one describes the failure of a software project to achieve its goals [86, 87]. One of the classical examples of the genre is William H. Whyte's groundbreaking

[3]Collins, *et al.* studied 'tacit' or implicit knowledge and the difficulty of transmitting what is not well understood even by practitioners [53].

study [262] of line and staff management of workers in the early nineteen-fifties, although Dalton's portrait of managers a few years later is somewhat closer to the method used in the present research [70].

3.6.1 The Form of Grounded Theory Used in this Study

Dalton was an early ethnographer of the Chicago school [22]. In light of modern criticism of the methodology of Dalton and his ilk as 'dangerous' and 'unethical' [22, pp. 123, 128–9, 132] it should be obvious that ethnomethodology has a complicated reputation. The goal of ethnomethodology is to discover the participants' worldview, but the success of that depends on whether the information the researcher wants to elicit is something that people are able and willing to talk about. Ironically, some of the best ethnomethodology has been done in nuclear weapons design laboratories, but the unique environments of Los Alamos and the Livermore laboratories in the nineteen-sixties had much to do with it [105]. Cameras and recorders are disallowed in SCIFs; barred from using the most important tools of standard ethnomethodology, the researcher turned to a related technique, also borrowed from sociology, known as Grounded Theory [99].

It has long been known that controlled experiments are extremely difficult to do in software engineering. Given typical project sizes, to find out whether a trial was successful can take years [150]. For comparison, the R' security evaluation consumed nearly 1.5 years and cost over a million dollars; the R'' DIACAP certification ran for fourteen months (including post-certification patches) and over eleven person-years of work.

Grounded theory methodology is ideally suited for the present purpose. It is an inductive methodology for discovering general theories in specific evidence (Figure 3.4). Above all, grounded theory is flexible [42, 201]. One

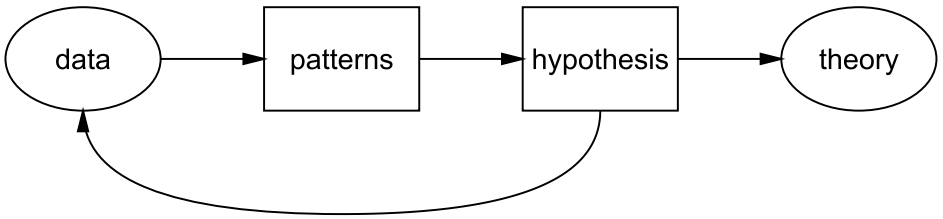


Figure 3.4: Grounded Theory methodology

way of looking at it is as a way of retrospectively running experiments on data already collected. Events, occurrences, personal names, places, words, phrases, even thoughts of the researcher as documented in analytical memoranda are coded (i.e., tagged) wherever found in the data. Codes are small mnemonic words or phrases suggesting the gist of an idea; supplementing the data, a database of codes is gradually assembled. As relations between codes are hypothesised or discovered, categories emerge. Categories shift, merge, split, and change as new data points are analysed. As more and more data are looked at in this way, eventually some categories may be found to be converging to one or more fixed points; these are considered prospective theories. Eventually, one or more theories is found that stops changing when tested against all available data. The theory or theories that emerge are grounded in the data, hence the name 'grounded theory'.

Personal recollection of classified meetings, reports, and conversations is thereby transformed from a handicap into an advantage; denied later access to verbatim reports or recordings, the researcher's unclassified notes and minutes of meetings were discovered to correspond approximately with the grounded theory practice of writing *analytical memoranda* [49]. Verbatim coding and *in situ* coding were impossible due to the indirect nature of the data; the version of grounded theory methodology used in this thesis therefore owes a debt to

the previous work of Melville Dalton and William H. Whyte that originally inspired Glaser and Strauss to formalise Dalton's method in the nineteen-sixties, more than it does precisely to the automated workflow of qualitative data analysis tools such as *ATLAS.ti*. The reason is because essentially all the data comprise analytical memoranda (in Glaser and Strauss's terminology) and none of it is directly traceable to the individuals being studied. Exactly as Melville Dalton worked in the organisations he studied, furtively hiding his notes from his research subjects, here the researcher was forced to type notes from memory after each meeting in the SCIF—twice, because the first (classified) version remained in the SCIF; then the researcher typed a new, releasable version of the notes (as soon as possible afterwards) on an unclassified computer. To the extent possible, without running afoul of security, the notes covered the same ground, but were necessarily a layer away. Weekly, the researcher digested the unclassified notes into an export-controlled *third* version for discussion with his supervisors. It is these third generation reports—several hundred in all—that comprise the raw data and analytical memoranda used in this study. It is argued, consequently, that the method is closer to Dalton's than to Glaser's [70, 'Appendix on Method'], and [71].

The goal of the grounded theory methodology used here is to understand how and why the interactions of cross domain solution developers, government information security certifiers, penetration testers, IV&V contractors, and the government Programme Management Office (PMO); the activity of regression testing (contracted by the certifiers to another department of P_1 in Charleston, South Carolina, but using the developer's own test procedures), accreditors, and data owners combine to determine progress through required activities in the standard process during the course of cross domain C&A,

and hence the cost in both time and money.

Specific Criticism of the Grounded Theory Methodology

Even prior to the invention of grounded theory, pioneers of participant observation knew that *'[the] practice, of course, moves from respectable to reprehensible shades. To some persons in the social sciences, it is both scientifically and ethically suspect. There are, indeed, several scientific shortcomings...'* [71, pp. 74–5]. It might be argued that the grounded theory methodology is unscientific because it formulates hypotheses after data; Biernacki makes the case for it in his 2012 book [27, Chapter 1]. On the contrary, in this case, we argue that yes, grounded theory *is* the reverse of the Popperian method, but uniquely it can be used to run experiments retrospectively when data are scarce and expensive and difficult to obtain. Such is the case in software engineering. The interpretation used here of the historical foundations of grounded theory methodology was the only way that an irreplaceable data set could be widely published.

3.6.2 Tool Support

A software licence for ATLAS.ti version 6.2.27–8 was purchased and the suitability of the tool to the available data was assessed. All of the data existed in the form of plain text notes in a single large file, ordered chronologically and time-stamped, but intermixed with unrelated data, analytical memoranda, and irrelevant other text. After importing the single large text file into ATLAS.ti and experimenting with coding it, that approach was abandoned. Besides audio and video files containing data, ATLAS.ti seems most suited to coding in relatively small PDF files, for which it has a good user interface and highlighting tools. The UNIX `csplit(1)` utility was used to extract a total of 817 time-stamped items from the electronic laboratory notebook file,

followed by `sed(1)` to add a LaTeX header with a unique sequence number to each data file for tracking purposes. [4] Items were converted to PDF for import into ATLAS.ti.

Coding

An initial manual sorting of the raw evidence collection was done to produce a set of likely codes to use. The initial set of codes was input to the Auto-Coding feature of ATLAS.ti and used together with regular expressions to relatively rapidly locate and tag every occurrence of the initial set of codes in the eight hundred PDF files. Proper names, project designations, locations, dates, mentions of the words 'schedule', 'finding', or 'board', testing phases, organisation names, and indicators of progress were further given to the tool until most of the PDF evidence files had been tagged in at least a few places. Variation in the number of codes assigned to each piece of evidence varied widely, but this is attributable to the equally wide variation in length, type, and density of the evidence files, which ranged from a single sentence to over a dozen pages of single-spaced text. The Auto-Coding process was deemed finished when every one of the *a priori* codes in the initial list was done, it could be observed that many of the auto-coded phrases were beginning to overlap in the data, and most of the evidence files had been visited. Coincidentally, the tool began to slow down noticeably around this time, indicating that it was reaching capacity limits.

Discovering the Categories

Concurrent with the accumulation of a starting set of codes during the initial manual overview of evidence, and whilst guiding the Auto-Coding

[4] GNU syntax is idiosyncratic; the `csplit(1)` command line used was actually `csplit -n 4 notes.tex '/^\\item\{[0-9]{8}\.[0-9]{4}\} /' '{*}'`

process, the first categories began to emerge. Ultimately, ten categories were found shared between the R' and R'' case studies (Appendix A). The same technique was not used for the R' case study, as the much older data existed in a completely different form.

3.6.3 Approach

A chronology of events in each case study was first prepared, along with an index of participants that was immediately used to generate the list of anonymisation codes. Organisation charts were obtained for the developer's organisation, D, and E, but approximate organisation charts could only be drawn for the validator's organisation, L, and it proved impossible to get any such information from the customer, C, the evaluator, N', the evaluator's technical adviser, N, or the customer's technical adviser, G. Because of shifting responsibilities, especially during the R'' certification process, organisation charts were inferior to induction of the actual power and influence relations amongst participants, which were coded manually by means of relations in the ATLAS.ti Network View Manager feature.

The methodology used here evolved from, at first, an analysis of a single engineering failure—the ill-fated Common Criteria evaluation of R'. This led to reading the literature of engineering failure, and to a number of red herrings and blind alleys along the way. When, late in the research on R', the opportunity to study a second, related case arose, the methodology was adapted to collect data from the R'' security certification process under DIACAP using NIST SP 800-53 security controls and the ICD 503–recommended portions of NIST SP 800-37 risk assessment process, in spite of the fact that the method of data collection and type of data collected were completely different in form, quantity, and quality from those available from the earlier

project.[5] Further, the outcome of the R'' certification was not known yet. Direct comparison, however, of two closely related variations of the same software, developed by the same people over a period of years encompassing several 'generations' of security accreditors, was too good a chance to pass up. Meeting minutes were exchanged for access.

A promising mathematical model was abandoned in 2011.[6] Two theories grounded in the data of R'' and R', respectively, one predicting certifier and accreditor behaviour during the CT&E phase of cross domain solution testing and one offering the developer a measure of control over accreditor behaviour in the ST&E phase of cross domain system testing, were discovered late in the process; they are described in §6.2.6 and §6.2.7.

The methodology evolved away from the original set of case studies and became increasingly general and more abstract as the grounded theories gained functionality and wider applicability. The research was no longer limited to description of particular cross domain solution products with particular features or particular accreditors at particular security levels, but developed a more powerful abstract accreditor model with weaker preconditions.

3.6.4 Refinement

The grounded theory was further developed in whiteboard sessions with the researcher's supervisor. Beginning from a set of mechanically generated network relationship diagrams, events in the certification and accreditation of R'' (and parallel events from R_0 and R', where applicable) were counted

[5] Data from the R' project consisted of emails, schedules, plans, many generations of draft documents comprising all of the evaluation evidence, tools, and validation reports. There were no meeting minutes at all, no analytical memoranda, and few quotations. The form of the data was completely different to R'' and very much inferior. Only a handful of documents survive from the even earlier R_0 project.

[6] The mathematical/physical model of accreditor interactions developed in 2010 is still promising, was anecdotally validated by several working accreditors, and may be revisited in the researcher's post-doctoral research.

and categorised into lists of explicit and implicit information flows—formal and informal communication channels—which were the linkages of the C&A mechanism. At no time were motives guessed or imputed to events, although statements of participants other than the researcher were fair game for interpretation. Formal channels were marked as 'formal' and checked off against a list of expected formal channels comprising deliverables in the DIACAP process [244], contract line items in the statement of work, and *ad hoc* reports that appeared as action items in meeting minutes. Informal channels, after cross-referencing against formal channels to verify they were not supposed to be there, were the primary focus of study. Why did they exist? What kind of information was carried by informal channels—was it different from formal channels? How much (qualitatively speaking) information travelled this way? Most importantly, was there a qualitative difference between the existence or non-existence of detectable informal channels—or between the relative effectiveness, or between the relative amount of information apparently being transferred across informal channels—respecting successful *vs* unsuccessful outcomes vis-à-vis the R'' and R' case studies?

3.6.5 Evolution of the Grounded Theory

Ultimately, three distinct populations were studied. The first population consisted of participants in the R'' case study: developers, certifiers, professional security testers, independent verification and validation contractors, and the sponsoring government programme office. The first population was large, but insufficiently complex because there was only a slight amount of need-to-know differences within it. The second population comprised the developer's installers and the set of accreditors they had encountered in the field while setting up R systems. This supplied a wide variety of ex-

amples, documented in the installation database, of inter-accreditor security clearance and classification level differences. The second population served to generalise the grounded theory to collateral classifications and make it deterministic. The third population was participants in the R' case study, because it included international accreditors; this helped prove that the grounded theory was complete.[7]

A surprising thing emerged from following this methodology: the failure of the R' security evaluation turned out not to be as interesting as earlier believed. The success of the R'' certification, correspondingly, was only another data point. The most interesting thing was the discovery of a general theory that, it is believed, might allow CDS developers to predict the behaviour of certifiers and accreditors and to exert a measure of control over the schedule of ST&E, both recurring activities that are important to their operations. To understand why, consider what happened in the case of R', when an experienced developer encountered a new security certification criteria for the first time.

[7] SCI accreditations are structurally equivalent to international accreditations, as will be shown in Chapter 6.

Chapter 4

The Unsuccessful Common Criteria Evaluation of the *R*-prime System

> *'A maxim of technology is that failures reveal underlying mechanism. A good way to learn how something works is to push it to failure. The way it fails will usually tell you a lot about how it works.'*
>
> —Trevor Blackwell, in the `analogtv.c` source code [29].

Foreign customers, whether government, military, or civil, would typically demand to see a Common Criteria certificate of evaluation of a product's information security level of assurance before they would buy the product, and that was a significant expense (at least several hundred thousand dollars) for the software developer to obtain.[1] The *R* developer, government contractors by nature, were loathe to invest that kind of money out of the company's

[1] Portions of this chapter were previously published in the *Second International Conference on Advances in System Testing and Validation Lifecycle* (VALID 2010), Nice, France, 22–7 August, 2010 [142].

own pocket. It would be much better, thought the developer, if a customer could be found to pay for the Common Criteria evaluation of R' itself; afterwards, the developer would benefit from the existence of that certificate, being able to use a Common Criteria certificate—along with a little creative interpretation about which version of the software it actually applied to—to sell R in an entirely new civil market[2] at home, as well as R' to foreign governments overseas.[3]

4.1 The Decision to Seek Common Criteria Evaluation

R by this time was solid and reliable 'old code'. Developed in response to urgent operational need beginning in 1992 after the U.S. military and intelligence community's experience with manual cross domain systems in the Gulf War, the R software had been in continuous development and maintenance for nearly fifteen years at the time of these events. It was a reliable, well-regarded and trusted cross domain solution that was accredited and running in hundreds of locations around the world—fixed, mobile, and shipboard—all of them classified.

D pioneered the concept, through the development of R, of the automated cross domain solution, the newness of which necessitated that extraordinary

[2]Civil applications of cross domain solutions envisioned by the developer included financial reporting systems, Electronic Health Record (EHR) systems, state and local law enforcement, and airport security. Because R was not well known in these application areas, D needed both a discriminator and a mark of software quality, which the Common Criteria certificate could fulfil.

[3]Technically, under the National Policy Governing the Acquisition of Information Assurance (IA) and IA-Enabled Information Technology Products (NSTISSP) number 11, the developer was already required to have a Common Criteria certificate to sell to the federal government, but compliance was grandfathered in for contracts signed before 2003, an exemption held to apply to R, even though typically a new contract was negotiated each year [56]. The developer felt some urgency to obtain a Common Criteria certificate for R to remain within the spirit of the law.

care be taken during the design and construction of the first versions of the R software, and carried through to every subsequent version, functionality upgrade, and maintenance release since. A machine had never been trusted with the 'release decision' before. The release decision, encoded in the form of 'rules' embedded in a 'configuration', would be the primary focus of accreditation testing later on. The software comprising the Trusted Computing Base (TCB) would be tested during certification. Because accidental release of high-side messages to a low-side recipient was thought to be the principal potential weakness of software running on commercially available hardware, a defence in depth philosophy was adopted, operationalised in the existence of a separate 'output guard' process designed into R. The output guard used an independent set of rules and served as a backup for the down-grader process, enforcing a minimal set of content and formatting criteria on messages before they reached an output channel process. The design mimicked the 'two-man rule' long in use for access control to highly valuable items like cash vaults in commercial banks or nuclear weapons in the military.

The software was designed and developed with careful attention given to reliability, audit, testing, and software development methodology. Originally a 'black' programme under an undisclosed government agency sponsor carrying the designation 'L-142', the programme underwent a number of changes of the government programme office having responsibility and budget authority for it across different branches of the U.S. military and intelligence community in the years since inception. This discontinuity of government programme office oversight is important, but belies the continuous scrutiny that R underwent, continuing to the present day, over the security and reliability of the software, its development environment, the vetting of software development personnel, control over software functionality, and security

testing (both certification and accreditation).

Being a cross domain solution, it can be understood that R had been tested many times by different authorities in diverse conditions, locations, and uses. The software was approved for use by the U.S. intelligence community starting in 1992–3 under a particular set of security testing criteria known as Director of Central Intelligence Directive (DCID) 6/3, and then by U.S. military branches individually under the Secret and Below Interoperability (SABI) rules, and for use in mixed SCI and collateral environments under the Top Secret and Below Interoperability (TSABI) rules, all of which are approval and accreditation criteria, not certification schemes.[4] Being the first of many automated cross domain solutions, R was specified and largely designed by the user community and evaluated by NSA and the Defence Information Systems Agency (DISA) before the unprecedentedly risky concept of an automated cross domain solution was finally accepted by the U.S. intelligence community in 1995.

For a new product (R') whose pedigree derived from one that had been evaluated successfully so many times in the past, it was thought by D that obtaining a Common Criteria certificate for R' would be relatively easy. The necessary security functionality was present, the software development environment already provided a high level of assurance of quality and test

[4]Terminology is important here; DCID 6/3 was not a certification programme, and NSA did not certify cross domain solutions under DCID 6/3; as it does today, NSA's Information Assurance Directorate (IAD) only provided a technical opinion. Each particular cross domain system containing R and installed in a particular location was accredited by one or more members of the intelligence community under the Director of Central Intelligence, and cross domain solutions, like R, could be 'approved for use' and placed on an Approved Products List by the Defence Information Assurance Security Accreditation Working Group (DSAWG). The Common Criteria and the Defence and Intelligence Community Information Assurance Certification and Accreditation Programme (DIACAP) are certification programmes. NSA, in conjunction with the National Institute of Standards and Technology (NIST) under the auspices of the National Information Assurance Partnership (NIAP) Common Criteria Evaluation and Certification Scheme (CCEVS), are the certification authority for Common Criteria security evaluations in the United States.

coverage and matched well the software development process maturity level called for, and after examination of the Common Criteria certification requirements, it was believed by D that the evaluation process would consist mainly of the developer exhibiting existing documentation to the evaluator. The researcher was assigned the task of writing the Security Target (ST) for the R' version of the software.[5]

4.2 Structure and Processes of a Common Criteria Security Evaluation

The 'evaluator' in a Common Criteria security evaluation is one of the certificate authorising members of the CCRA. The evaluator is chosen by the developer, often in consultation with the validator, and in many cases, the intended customer. In the case of R', although the customer C was not the United States government, the chosen evaluator was N, the U.S. scheme. It is not uncommon for vendors to select an evaluator outside of their own country, either for reasons of national loyalty, or evaluator backlog, or out of a perception that some evaluators are more or less permissive in their security testing diligence than others. 'Evaluator shopping' is a legitimate criticism of the Common Criteria as discussed in section 2.5.2. The reasons for choosing evaluator N in this case are unknown, but N's reputation is that of being a strict evaluator.

[5]The author of this thesis had assisted with an earlier project (designated R_0) in 1999 whose goal was the preparation of a complete package of 'evaluatable evidence' for an earlier version of R for the M project. The ST and design documentation for R_0 turned out not to be very useful to the preparation of the work packages for R', however, because of differences in functionality between the Target of Evaluation (TOE) of R_0 and R', some new functionality included in R', some existing functionality of R that was not included in R', and the fact of a completely different installation environment between the two. The same author, tasked with writing the R' ST in 2004, was however able to use some of the experience gained writing an ST under the earlier guidance of an experienced ST writer during the 1999 project.

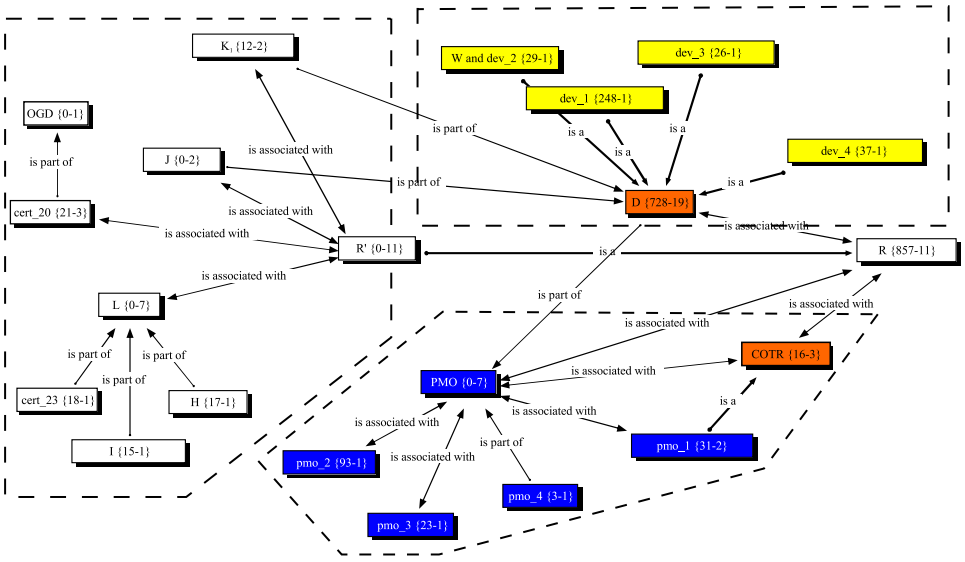


Figure 4.1: J and K were the primary interface between D and L; in accordance with the Common Evaluation Methodology (CEM), there was no direct communication between D and the certifier, N. OGD briefly interacted with D through cert_20 but dev_1–4 were working on R and un-involved with R'. The Programme Management Office (PMO) played a hands-off role.

The 'validator' in a Common Criteria security evaluation is one of a small number of approved Common Criteria Testing Laboratory (CCTL) outfits, which are commercial enterprises whose purpose is to work with the developer through the writing of an evaluation documentation package, making sure that everything is complete and up to standard before the evaluation package is seen by the evaluator. Some of the national schemes, including N, do not accept evaluation requests 'over the transom'; these must come in through an approved CCTL.[6] The validator assists the developer with preparation

[6]It is interesting to look at the validator–evaluator relationship by analogy to dentists. Specialist oral surgeons (the national scheme evaluators) rarely encounter teeth covered in calculus and plaque, because the specialists' patients are referred to them by general practitioners (the validators) who take care of minor dental health care problems and teeth cleaning before the specialist ever sees the patient. There is a risk that this might lead specialists to underestimate the incidence in the population of minor dental health problems, in much the same way as Common Criteria evaluators encounter only fully

of work packages. Some validators have a process in mind as to the level of scheduling detail and format of the Work Breakdown Structure (WBS), and usually assign a technical representative to work with the developer. In the case of R', the validator was the corporation L, the validator's technical representative was the individual H, the validator's project manager was the individual I, and the developer's technical representative was the individual J. Responsibility for project management in the developer's organisation shifted over time from one individual to another collectively designated here as K_1 and K_2. (See Figure 4.1.)

The ST is the top-level introductory and organising document of a Common Criteria security evaluation and heads up a large collection of evidence that a developer provides to the validator. The ST describes the Target of Evaluation (TOE) and specifies which particular SFRs and SARs are claimed to be satisfied by the TOE. The EAL determines the SARs to be evaluated and the Protection Profile (PP), ideally, determines the SFRs.

Oftentimes an ST is written with reference to an existing PP—originally a predefined set of SFRs and SARs determined by and for a particular type of end-user and published for the benefit of the community—but in the case of the R' evaluation no suitable PP for cross domain solutions existed at the time, so the developer was free to select a collection of SFRs that well represented the total capabilities of R'.[7] This ability to proceed without a validated PP was a doubled-edged sword, and has since been curtailed by evaluator action; on 16th March 2010 the N Interpretations Board announced that it had '... determined that the current U.S. Protection Profile robustness

validated work packages, leading the evaluators to overestimate the level of security and quality in commercial products.

[7]The developer is allowed to choose which version of the Common Criteria to use; the version is specified in the evaluation report. At the time of the R' validation, version 3 of the Common Criteria was not yet standardised, so the developer, in consultation with the validator, decided to use version 2.3.

model needs to be revised' and that henceforth a validated PP would be required [165].

On the one hand, the freedom of the developer to select the SFRs to be evaluated might allow an unscrupulous developer to cherry-pick a set of SFRs that are easy for the developer to satisfy, while ignoring other SFRs that represent important security functionality that a customer—who depends on the CC to identify products with good security functionality—needs to have. But the other consequence of freedom to choose SFRs can negatively affect a naïve developer, as may have happened in the case of R'. The absence of a published PP for cross domain solutions likely led D to include too many SFRs in the ST, which led directly to problems in circumscribing the TOE boundary and inflated the cost of the validation (see Chapter 6).

The evaluation evidence for a product is divided into work packages. The number and type of work packages needed for validation depends completely on the Evaluation Assurance Level (EAL) being claimed. The EAL of the R' evaluation was set at 'EAL4+', meaning it included all the SARs for EAL4 in Table 6 of Part 3 of the CC with the addition of ALC_FLR.2 Flaw Reporting Procedures which were not required at EAL4 in version 2.3 of the CC. [57]. A minimum level of EAL4 was required by the customer, and as EAL5 would have been excessively expensive for the purpose, in the end, the developer requested that EAL4+ be attempted, again because EAL4+ matched most closely with the R' software's native level of assurance in the software development environment.

The validation and evaluation process normally proceeds from the traditional project kick-off meeting through an interval of several months where the developer works closely with the validator to produce a defined set of work packages. The validator is provided with an installation of the TOE.

At the time of the R' validation, the Common Evaluation Methodology (CEM) of the CC version 3 had not yet been promulgated, so the validation methodology employed was proprietary to L [59]. In the course of events, the R' product was listed on N's web site as 'in evaluation'. Had the validation process of R' completed as planned, L eventually would have delivered the work packages to N for evaluation. The evaluation process would have proceeded to testing of all SFRs, review and inspection of all of the documentation associated with SARs, writing of an evaluation and certification report, and issuance of a Common Criteria certificate.

4.3 Chronology of Events in the R-prime Evaluation

Writing of the ST for R' began shortly before the selection of the testing laboratory L was finalised, as it was believed early on in the project that the previous R_0 ST, despite never formally having been validated or evaluated, could, with minimal changes, be used as the template for an updated ST for R'. While that belief was true in principle, differences in the feature set between R_0 and R' caused by addition of new features in the five years that had elapsed since the preparation of the first ST, obsolescence of some functionality followed by its removal for dead code reasons (mandated by Q, the approver of R''), selective elimination of certain advanced functionality of R kept outside the R' software development configuration management system branch for export control reasons, and changes to the Common Criteria from version 2.1 to version 2.3, made that aspiration moot; in the end, the template for the R' ST was generated automatically from machine-readable copies of Part 2 and Part 3 of the Common Criteria after they were

mechanically transformed from PDF to HTML. Automatic generation was the driving methodology on D's end of the project throughout—wherever possible, to generate new documentation automatically from existing primary documentation or source code from the version control repository—and to re-use existing documentation unchanged as much as possible.

Setting the TOE boundary correctly in an ST is an art [184]. It was a significant problem in the R' evaluation. If made too small, the cost of validation is reduced but the evaluator may baulk that not enough security functionality is tested. If made too large, the cost of validation and the time required for evaluation are both increased. The primary difficulty lies in two directions. Firstly, if the TOE, like R', is an extremely configurable device with many capabilities, then by rights the ST ought to be comprehensive about the TOE's full capabilities, even if making it so would balloon the number of SFRs to an unmanageable size (an oversized TOE cannot be said to be an un-testable size, because the developer's own software development process must have been able to cope with testing it; the issue at hand is only the cost of the evaluator's time in re-testing it, but see §6.3 for a critique of how this is done in practice). The real problem with increasing the number of SFR claims in the ST is that it inflates the size of the design documentation problem.

Secondly—and typical of a software product that, like R', was designed to run on Commercial Off-the-Shelf (COTS) hardware—there is the important question of whether the TOE includes COTS hardware, software, and firmware. If the TOE incorporates COTS hardware components, it can open the developer to the risk of having to justify security-relevant hardware design decisions to an evaluator when the developer had no involvement in those choices. An example is the provision of a feature called Lights-Out

Management (LOM) on certain Sun Microsystems machines, comprising a de facto additional network interface that D was obligated to disable or physically remove before deployment, even though the manufacturer of the hardware assured it could not be used to ex-filtrate data; D appreciated the nature, inclination, and capabilities of the security testers at N' and G better than the hardware manufacturer, and risked invalidating the manufacturer's service contract by physically removing these components even though the operating system had no documented or obvious undocumented means for accessing the LOM network interface after boot.

Incidentally, although not precisely at issue in the R' evaluation because S was to be a one-off system—but highly relevant to the anticipated wider applicability of an R' CC certificate to R—hardware vendor life-cycles are a continually renewing problem to cross domain solution developers. Security evaluators, CC and otherwise, rightly regard the physical hardware on which a cross domain system runs as being within range of attack during analytical risk analysis or penetration testing, because the expected adversary logically will explore all opportunities looking for an exploitable vulnerability. Certification authorities and standards writers are aware of this and typically make security certifications dependent upon specified hardware and operating system patch levels and mandate that changes to hardware are treated as software version changes would be under the certification rules. The difficulty arises because COTS hardware manufacturers, bound by upstream component manufacturers, who are in turn bound by upstream semiconductor process changes in the supply chain, which itself is forced by the extremely high capital cost of steppers, advanced light sources, fabs, and optics to keep up with Moore's Law, regularly stop producing old hardware and introduce new machines. For most end-users, with less stringent security

requirements, this is not a problem; hardware manufacturers provide pre-release hardware to operating system developers, and operating system versions and patches are kept in step with annual (or at some other regular interval) hardware introduction cycles. But cross domain solutions are almost invariably developed for special high-assurance COTS operating systems, which are themselves certified, and the certification life-cycle for certified operating systems is both out of sync and out of phase with the hardware vendor life-cycle. Specifically, hardware life-cycles for general-purpose COTS computers range from six to eighteen months, phase-locked to semiconductor process manufacturing as described above. But certified operating system release cycles—invariably, since the replacement of ITSEC and TCSEC by the Common Criteria—are tied to the duration of Common Criteria evaluation, and because operating systems are usually large and complex software systems, CC evaluation of them always takes more than a year. The result is that certified operating system versions sometimes run only on obsolete and sometimes even unavailable hardware.

This happened several times to R's developer. Two divisions of a particular computer and operating system vendor, one division responsible for building hardware and the other for writing and certifying a specialised and certified version of the operating system with additional functionality to run on the same company's hardware, did not talk to one another and fell out of sync for nearly a year. New hardware was introduced that was incapable of running the old operating system. The conventional, non-certified operating system was updated accordingly, and patches released for customers who preferred to continue running older versions of the operating system. The problem, for D, was that the new operating system was not yet Common Criteria certified, and some of the patches were inapplicable to the certified

operating system, but regardless, all of the operating system patches were un-allowable to be applied to the certified operating system because the certification was written solely to the patch level of the operating system TOE that had been evaluated. If it had ended there, the situation would have been familiar to *D* and to *D*'s customers, if less than desirable. Sites running classified cross domain systems are regrettably used to running old, un-maintained hardware and operating systems without the latest security patches for precisely this reason; it is not a good situation, but site security officers and accreditors tell themselves that physical security, relatively isolated networks, and increased levels of vigilance to intrusion detection systems and intrusion protection measures like screening routers will compensate to some degree for the absence of up-to-date security patches. However, the hardware manufacturer immediately stopped making the older computers on which the certified operating system depended. The *R* developer was forced to search urgently for alternative sources of supply, and for reasons of supply chain security, most second-hand sources could not be used. *D* was ultimately forced, with the assistance of the U.S. Navy, to mine ships at sea for hardware-in-use that had acceptable pedigrees in order to meet contractual obligations for cross domain system installations. This happened three times between 1998 and 2006, for three different models of hardware and two versions of the certified operating system.[8]

The world of classified cross domain systems is not the only place where the same thing happens. The difference between the time it takes to design new COTS hardware—which is typically never certified except for Network Equipment Building System (NEBS) Level 3 and MIL-STD 810 for en-

[8]The affected hardware included Sun Microsystems Inc. Ultra 5, Blade 150, and Blade 2000 computers, with Trusted Solaris 2.5.1 (TCSEC certified) and Trusted Solaris 8 Certified Edition (which was Common Criteria evaluated) operating systems.

vironmental conditions and ruggedness, FIPS 140-2 for certain specialised cryptographic hardware modules used in unclassified applications, or NSA Type 1 for classified cryptographic hardware—is much less than the time required to certify a major operating system release. A similar situation occurs with medical diagnostic and therapeutic devices. In the U.S., the Centre for Devices and Radiological Health (CDRH), a branch of the Food and Drug Administration (FDA), controls the approval of medical devices. CDRH regulations, not unlike Common Criteria security evaluations, require Class 3 devices to undergo clinical trials and design reviews before approval, an expensive and time-consuming process. Manufacturers of Class 3 devices are understandably reluctant to jeopardise the approval of devices installed in hospitals and clinics, and in the case of medical devices incorporating COTS hardware and operating systems such as Microsoft Windows, reportedly prohibit or discourage buyers from installing unapproved security patches for fear of triggering a re-certification event [230]. A contributing factor to security patch–non-compliance in cross domain systems installed and operational in very remote locations is the expense of travelling to sometimes war-torn regions and the necessity of disrupting critical communications for hours. Safety-critical airborne and spacecraft systems have their own peculiar concerns around catastrophic failure and inaccessibility for repair, and deal with it in their own way, deemed acceptable for these applications, by means of testing and use of redundancy [197, 109].

4.3.1 Work Packages

Part 3 of the Common Criteria, through the mechanism of requiring certain SARs for evaluation, largely determines the size and composition of the written materials required to be submitted for validation, called *work packages*.

SFRs are not documented, particularly, except by reference in the ST to particular strength of function claims, and are expected to be verified by the evaluator, for the most part, by exercising the TOE and through examination of design documentation. Work packages were delivered incrementally from D to L during the validation process according to the project schedule under the control of project managers K_1 and K_2.

In the case of R', most of the SARs in Table 4.1 were already completely satisfied and adequately documented. For example, the original software development methodology required and specified by Intelligence Community (IC) members at the beginning of the R project naturally included an automated Configuration Management (CM) system, satisfying ACM_AUT.2 (although ACM_AUT.1 was claimed) with more than enough capabilities and scope to satisfy the requirements of ACM_CAP.4 and ACM_SCP.2. The developer's installation procedures, unchanged, were sufficient to satisfy ADO_IGS.1; as described previously in the discussion of hardware supply chain security, D found it necessary to document a secure delivery process in order to satisfy ADO_DEL.2.

The R' *User Guide* and *Trusted Facility Manual (TFM)* easily satisfied the requirements of SARs AGD_ADM.1 and AGD_USR.1 for user and administrator guidance, respectively. The developer's process already generated specific editions of both of these books for each release of the software, so the Documentation and Training department had them available as a matter of course. Because of the configurable nature of the system, the *R User's Guide* is both an administrator's and operator's guide, and the TFM is a guide for site security officers, fitting into the category of a traditional administrator's and security features configuration guide.

In the 'Life cycle support' category, the developer supplied documentation

Assurance Class	**Assurance Family**	**Components**
Configuration management	CM automation (ACM_AUT)	ACM_AUT.1
	CM capabilities (ACM_CAP)	ACM_CAP.4
	CM scope (ACM_SCP)	ACM_SCP.2
Delivery and operation	Delivery (ADO_DEL)	ADO_DEL.2
	Installation, generation and start-up (ADO_IGS)	ADO_IGS.1
Development	Functional specification (ADV_FSP)	ADV_FSP.2
	High-level design (ADV_HLD)	ADV_HLD.2
	Implementation representation (ADV_IMP)	ADV_IMP.1
	Low-level design (ADV_LLD)	ADV_LLD.1
	Representation correspondence (ADV_RCR)	ADV_RCR.1
	Security policy modelling (ADV_SPM)	ADV_SPM.1
Guidance documents	Administrator guidance (AGD_ADM)	AGD_ADM.1
	User guidance (AGD_USR)	AGD_USR.1
Life cycle support	Development security (ALC_DVS)	ALC_DVS.1
	Flaw remediation (ALC_FLR)	**ALC_FLR.2**
	Life cycle definition (ALC_LCD)	ALC_LCD.1
	Tools and techniques (ALC_TAT)	ALC_TAT.1
Tests	Coverage (ATE_COV)	ATE_COV.2
	Depth (ATE_DPT)	ATE_DPT.1
	Functional tests (ATE_FUN)	ATE_FUN.1
	Independent testing (ATE_IND)	ATE_IND.2
Vulnerability assessment	Misuse (AVA_MSU)	AVA_MSU.2
	Strength of TOE security functions (AVA_SOF)	AVA_SOF.1
	Vulnerability analysis (AVA_VLA)	AVA_VLA.2

Table 4.1: Security Assurance Requirements for EAL4+ (after [57, Table 6]) with augmentations for R' in **boldface**

showing that R spaces were a DoD accredited Closed Area under the National Information Security Programme Operating Manual (NISPOM) and all personnel in the closed area positively vetted, satisfying ALC_DVS.1. The software life-cycle (an iterative spiral model), tools and techniques (such as coding standards and software inspection procedures, developer training, and review boards) were fully documented in the R Software Development Plan (SDP) which was supplied as-is in support of ALC_TAT.1 and ALC_LCD.1. 'Flaw remediation', the only augmentation above EAL4 that was claimed (although D could have claimed ACM_AUT.2 based on capabilities and usage if they had wanted), was satisfied by the developer's bug tracking database, which was integrated with the CM and build system, review boards, and software inspection processes as documented in the SDP. The developer decided, based on its recent ISO 9000 evaluation and Software Capability Maturity Model (CMM) evaluation at Capability Maturity Model Integration (CMMI) Level 3, that ALC_FLR.2 ('Flaw reporting procedures') was the most correct SAR to claim; ALC_FLR.3 ('Systematic flaw remediation') would be more appropriate for a CMMI Level 4 software development organisation to have.

The developer's existing test procedures and Test Plan (TP) for R' were supplied to L in support of test coverage (ATE_COV.2) and depth (ATE_DPT.1). Basic functional testing (ATE_FUN.1) was demonstrated by a Requirements Traceability Matrix (RTM) and the Configuration Item (CI) list, which linked every enhancement or maintenance Change Request (CR) to a Configuration Review Board (CRB) meeting, software inspection report, CM build, test procedure, test plan, and test report. Independent Verification and Validation (IV&V) was a required part of D's SDP for certified versions of R, so it was no difficulty to demonstrate compliance with

ATE_IND.2 requirements.

Some SARs were addressed in the body of the ST. Both AVA_MSU.2 ('Misuse') and AVA_SOF.1 ('Strength of TOE security functions') occupied many pages of exposition in the ST on such topics as password complexity, internal protection of the Identification and Authorisation (I&A) database—which had three layers of protection: Discretionary Access Control (DAC), Mandatory Access Control (MAC), and cryptographic functionality; and process separation, again using the facilities of the Trusted Solaris operating system for mandatory access control. In general, every security function in the R' application software was independently implemented, so far as was possible, by application code and operating system MAC configuration both, and auditing was similarly duplicated, although some of its work was distributed to one subsystem or the other based on information classification, by both application auditing code and operating system audit functionality. In support of AVA_VLA.2, a completely new vulnerability analysis was written for the appendix of the ST.

It was the 'Development' assurance class that gave D the most problems with the Common Criteria evaluation of R'. For historical reasons tied primarily to government funding and the previously mentioned 'orphan' status of the R programme with respect to Government Programme Office (GPO) tenure dating from the earliest days of the L-142 programme, the original R design documentation, while excellent, had been poorly maintained. This showed up when R' attempted to exhibit High Level Design (HLD) and Low Level Design (LLD) documentation for the ADV_HLD.2 and ADV_LLD.1 work packages.

Even supplying the original HLD and LLD for R in place of R' was not possible, for two reasons. Firstly, some of the original documentation had

been lost. Much of it never existed except the form of paper, some of it was thousands of miles away in California or Washington, D.C., and no one in the government programme office—through several changes of responsibility from O to F to P_1—knew where it all was. D had only fragments of the HLD and LLD, and those were many major versions of the software out of date. Secondly, as described in the context of re-use of the R_0 ST in §4.3, significant functionality differences existed between R and R' as they existed in 2006 and R in 1992. For the simplest example, communication interfaces in 1992 were RS-232 serial; in 2006 the serial ports still existed but they had been joined by TCP/IP over 100Base-T Ethernet and other protocols, data link layers, and physical layer interfaces. The rules engine had been completely replaced.

After searching the Software Development Library (SDL) and asking other software developers for their oldest paper files, the ST author, J, was forced to conclude that, given the time and budget constraints driving the ST and ADV work packages, which were on the critical path for validation, the only way to generate an HLD and LLD in time was automatically or semi-automatically. To that end, J investigated two parallel approaches. The free open source tool *Doxygen* was obtained and used in an attempt to generate a usable high level design documents from the R' source code in the CM repository [247]. At the same time, J wrote a series of complex UNIX `sed` scripts to filter the entire C++ source code of R' into fully hyper-linked HTML. It was hoped that the combination of class hierarchy diagrams produced by *Doxygen* for a high level guide with hyperlinks into the web browsable source code database would serve as an adequate substitute for the missing HLD and LLD documents. The validator's technical representative, H, was noncommittal. H preferred to see design documentation in the

form of Functional Flow Block Diagrams (FFBD) and attempted in several meetings to teach J how to produce them, but with no tool support for drawing FFBDs except by hand, and with much of the necessary information about data flows existing only in the head of the individuals W and T, few FFBDs were ever successfully produced. The author feels personally responsible for these shortcomings.

The necessity of automatically generating ersatz HLD and LLD documentation for R' was couched earlier as a philosophy, but it was a philosophy borne of desperation. In the absence of original and continuously maintained FFBDs that never existed for any version of R, that philosophy mismatched with L's work processes and probably delayed validation by several months. In contrast, the ST, although large and complicated and difficult to write, was successfully done. The automatically generated HTML source code browser was delivered to the validator in satisfaction of SARs ADV_IMP.1 ('Implementation representation') and ADV_RCR.1 ('Representation correspondence') and a section in the ST based on several old R Security Policy (SP) documents served for ADV_SPM.1. The last work package scheduled to be delivered, ADV_FSP.2 was still in progress when the R' validation came to an abrupt end.

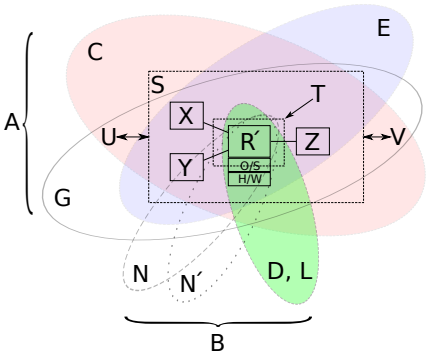
4.3.2 Setting the TOE Boundary

The author of the ST, J, was continually urged by the project manager K_1 to keep the TOE as lean as possible. In the end, the R' TOE boundary was established to include certain specified COTS hardware and software, but only to the extent of requiring particular vendor, product, version, release, patch level, and end-user configuration as specified by the developer [140]. Since few of the SFRs directly necessitated testing hardware functionality, and

since the developer already extensively modified and specifically configured the COTS Operating System (OS), the only real added work required of the developer was to establish and document a secure installation process for the COTS hardware and software, which was done and added as a new work package.

The fact that D included SFRs for all of the security functionality of R' in the ST, instead of minimising the set to claim only functionality needed by S, combined with the developer's willingness to accept N as the evaluator, supports the conclusion that D was more interested in upholding the spirit of the Common Criteria than of gaming the rules for personal gain. It is further supported by the fact that D was very familiar with and comfortable working with N' as the security accreditation authority in previous projects.

In retrospect, it is unclear whether the TOE boundary could have been set differently—*i.e.*, a smaller TOE—and remain within the ethical boundaries of good engineering practice. As a parallel example, consider the CC evaluation report for PR/SM LPAR for the IBM eServer zSeries z890 and z990 mainframes [118]. That TOE was tightly constrained, the ST and certification report both stipulating assumptions considered by D to be outside the realm of possibility for R', such as manufacturing origination of the hardware, and limitation of the TOE to a strict separation virtual machine monitor, completely disregarding the inter-partition information sharing capabilities of PR/SM LPAR [79]. The author of the R' ST believed it would be dishonest to limit the TOE only to that functionality used by S, since R' by design is a configurable cross domain solution with several types of general purpose communication channels and rules-based message handling that runs on limited and specified but commercially available hardware manufactured by a third party. Consequently, the ST for R' described the complete scope



A	Participants in country A
B	Participants in country B
C	Customer
D	Software developer
E	Systems integrator
G	A's government security agency
L	CC testing Laboratory
N	B's government security agency for Common Criteria (CC)
N'	Another government information-security certifying agency of B
R'	The product being CC-evaluated
S	System being developed by E
T	TOE boundary
U, V	Other systems outside S
X, Y, Z	Sub-components of S

Figure 4.2: Software developer D had influence only over the Target of Evaluation (TOE) R', whereas the system S as a whole was built by E. When the customer's government security representative G did their analysis, they considered the superset of S^* including the external interfaces to U and V. (Adapted from [142], after [238, Ch. 7].)

of functionality of the software and hardware to be delivered with S and the EAL (with augmentations) claimed in the ST matched D's software development process maturity level. Anecdotally, the R' ST was informally praised by N' in a meeting with D that was attended by J in which it was said that the R' ST, although larger than most, was appropriately complete in the details and disclosed precisely the kind of information that an evaluator needed to have.

It is interesting to compare the experience of Hill, *et al.* with the R'

validation experience, because that paper clearly shows the back-and-forth negotiation that occurs between the developer and the testing laboratory during the validation process, even showing an appeal to the evaluator to resolve an impasse [113]. Reading between the lines of that paper, it seems that some of the same discussions and negotiations occurred between their developer, validator, and evaluator as took place between D and L during the R' project, as in the amount of detail to be included in the ST, LLD, and HLD work packages. There is ever-present tension between developer and evaluator during a Common Criteria security evaluation—which expresses in terms of the size of the ST—over how much functionality is security-relevant and ought to be tested. It is in the developer's interest to minimise the TCB, and hence the scope of the TOE, in order to speed the process and achieve certification sooner.[9] It is probably in the evaluator's interest, and almost certainly in the validator's interest (so long as the validator is engaged by the developer on a cost-plus-fixed-fee contract) to maximise the scope of the evaluation, and hence fees. This is one of the most important purposes for having and requiring protection profiles; it fixes the SFRs and SARs for a given EAL, ensuring fairness within and across evaluations.

4.3.3 Asynchronous Customer Security Review

It is believed likely that the R' validation would have succeeded had the last work packages and the final version of the ST—the ST was continually being updated as served as index and glossary to the work packages—been delivered to L as scheduled.[10] The validator had been working with D throughout

[9] Smart card components, including integrated circuits and PIN pads—being in fact cryptographic terminals as much as their classified brethren—have equally small TCBs, so are typically evaluated to EAL5+ to comply with EMVCo and PCI standards [61, §7.3.3].

[10] It is logical to conclude that had the validation been successful, the evaluation would have proceeded normally, as that is both the purpose of, and the contracted goods for, licenced evaluation laboratories such as L.

the project, examining each work package as it was submitted by D, giving feedback on work packages, and tracking the validation schedule. With less than a week to go before the planned date for submission of the last two remaining work packages, the entire effort was unexpectedly torpedoed. The customer, C, had, without notifying D or L, engaged a disinterested third party, G to look at E's entire system design for S and render a technical opinion. This was C's prerogative. Figure 4.2 shows the relationship between C, D, E, G, L, N', and S. G pronounced S 'not fit for purpose' and C immediately terminated the project.

The individuals J and K_1 had both travelled several times to country A and had met several people in G. The researcher was very favourably impressed by the technical prowess evident in G and spent many enjoyable hours sparring with them about computer security attack and defence. G's final report, however, was as damning as it came as a complete surprise to D; it is not known whether the report came as a surprise to E as well. The report is classified and cannot be reproduced here, but G's main objection was not to R'—although it did come in for some specific criticism,[11]—but to an unrelated subsystem of S that had been designed and implemented by E. The developer of R' had never seen the components X, Y, and Z of the larger system S except as boxes on a block diagram,[12] but G recommended strongly against the deployment of S in the field.[13] For a short time after cancellation of the project, E threatened to bring a lawsuit against D for

[11] D took to heart the specific criticism of G with respect to R'—and by extension, R—and immediately implemented an expensive, uncommon, and only recently become technically feasible improvement to D's software development process. It was successful.

[12] S is the system designed by E that contained R', which was designed by D, in addition to components X, Y, and Z supplied by E. The larger system S^*, including its external interfaces to outside systems U and V, was the purview of C and is the system that was examined by G.

[13] One of the reasons given by G for the recommendation to cancel S was the presence of un-patched Microsoft Windows 2000 servers in X and Y.

breach of contract, answering a question once asked by Andrew Tobias in 1970, when he

> '... wonder[ed] whether one subsidiary of a conglomerate has ever sued another?' [233, p. 113].

4.4 Post-Mortem

A contributing factor to the failure of the R' security evaluation was turnover and tenure of project managers. Over the course of L's validation, project managers changed at least six times, and in both the D and L organisations.

Turnover amongst project managers (PM) has been repeatedly implicated in software and other project failures [39, 40, 203, 109, 194, 155]. Parker and Skitmore (2005) studied the effect of changes in project manager on projects and identified 'arrival effects' (from the point of view of the project manager) as primarily slowing progress after the arrival of a new project manager,[14] but 'project effects' (from the point of view of insiders on the project) are probably more important [174]. A possible mechanism to explain project effects is given in the next chapter.

But which way did causality run? Did the large number of project manager changes that were observed in R' cause the failure of the project, or were they caused by it?

> Our respondents singled out staff turnover as being a major contributor to both the success and failure of projects. In the successful projects there was either no staff turnover, or very little staff turnover [252, p. 1025].

Initially, it was thought to the former, but that fails to explain why

[14] 'Arrival effects' had been noticed earlier by Grusky (1963) and examined further by Gamson and Scotch (1964) but Grusky's focus was on the vicious cycle exhibited by repeated changes of baseball coaches in losing teams [93].

almost the same incidence of PM changes was seen on both sides of the certification, in two different organisations (D and L), with no direct connection between them. The changes of PM were not noticeably correlated between organisations over time. Grusky tried to argue something similar in 1963 but Gamson showed his postulated effect to be ambiguous. If it is assumed instead that project manager turnover was an effect, then we could argue that it was caused by, firstly, poor support of the failed project by the developer's organisation; secondly, lack of prior experience (on the developer's side) with the activity; and thirdly, indifference on the validator's side (contracted as they were on a cost plus basis), which led to repeated project manager changes because the project manager is always in possession of the best information. So project manager tenure is an indicator, not a cause, of failures.

This, then, appears to be the explanation for what Brooks, Sauer, Harland, Rost, and the National Audit Office found.

In the following chapter, it is proposed that changes in project manager disrupt 'implicit communication channels' necessary to the operation of a project.

Chapter 5

The DIACAP Certification of the *R*-double-prime System

> *'[The process] was like the solving of an algebraic equation. First you alter one variable, then another, but it never occurs to you that the constants might be wrong'* [54, p. 449].

The developer spent relatively little time licking its wounds after the unsuccessful outcome—both certification- and installation-wise[1]—of the R' project. The developer, D, routinely maintained dozens of projects concurrently, in all stages of progress from RFI to accreditation and maintenance, as well as an active programme of software development for enhancements. By early 2010, as a direct result of the hardware and OS vendor's life cycle road-map, it was recognised by D and the government Programme Office P_1 jointly that R needed to be ported to the current version of the certified multi-level operating system or else the ability for D to purchase new hardware for future installations would be jeopardised. The new OS port

[1] Because some Requests for Information (RFI) never reach the contract stage, and because engineers occasionally recommend a different solution, and because sometimes customers run out of funding before the end of a project, the developer's key metric is *installations* of R, defined as operational sites that are accredited and under maintenance.

would be called version R'' and would require a complete re-certification of the software.

5.1 Beginning of the Project

D faced a situation not unfamiliar to cross domain solution developers: once again, the rules had changed.[2] Despite that fact, new installations of R were occurring as often as ever, nothing untoward had happened to D's reputation as a result of the failure to achieve Common Criteria certification for R', and the developer was getting ready to begin the certification testing of R''. The new system would be an enhanced version of R with new capabilities, and at the same time would be ported to a new hardware architecture from a different manufacturer—one that D hoped would not fall out of sync again with the Common Criteria evaluated version of the COTS operating system underlying the R application software (a prerequisite of NSTISSP No. 11 for certification *and* accreditation, and one that was becoming increasingly difficult for D and the Programme Office to rationalise away).

But other changes were occurring in the Department of Defence and the intelligence community at the same time. At long last, DCID 6/3 had been retired and the IC and DoD finally had a common C&A standard for cross domain systems. DIACAP had been in use elsewhere for unclassified information systems within DoD for a few years, in accordance with government-wide FISMA requirements,[3] but in the summer of 2009, DIACAP was combined with Revision 3 of the NIST SP 800-53 security controls and the NIST SP 800-37 risk management framework, under ICD 503 (in the

[2] Recall the end of § 1.5 and Chapter 4.

[3] ‘ “In theory, [FISMA] is a road map for developing a strong cybersecurity program[me] at each agency,” [Daniel] Castro said. “In practice, agencies often focus on achieving compliance rather than effectiveness.” ’ [239, p. 33].

case of Intelligence Community systems), to serve as a single C&A standard for cross domain solutions and cross domain systems intended to handle information ranging from collateral to SCI. The problem was that no one had ever conducted such a certification before.

The situation was comfortably familiar to D. What made it interesting this time was that the certifier, Q, was equally unfamiliar with the new security testing criteria. It was decided by Q, in the person of the Acting Director of UCDMO, that R'' would be the first cross domain solution to be certified and accredited under ICD 503 with the NIST SP 800-53 security controls. On 11th November 2009, D hosted a kick-off meeting with Q, the programme office, and P_2 (the IV&V contractor), to set the ground rules for certification testing of R''.

5.2 Themes Identifiable from Observations

Preceding events served to sensitise the researcher, if not the participants at the time, to certain features of the software development environment of R that seemed to be important, even if their precise significance was not yet clear. Specifically, the difference between the outcomes of the two projects described here and in the previous chapter cannot be separated from the fact that many of the same participants around R' and R'' worked on both. The number of years of experience on-programme of the core software developers exceeded ten years each in almost all cases (T, W, dev_1 and dev_3, primarily); the certification authorities—with the exception of the government's professional security testers, who tended to be younger—and Programme Office managers,[4] as well as IV&V personnel were all 'native' to

[4]COTRs in charge of R's government programme office had a typical tenure of more than five years each, and at least twice a member of P_1's staff moved to D's organisation to take over as programme manager. Turnover of programme managers was equally low; software

the R programme and most had been involved (and in their current roles) since the programme's inception. The most obvious difference between R' and R'' is in the tenure of project manager (PM) personnel, echoing the concerns of Rost and Glass [194] and Harland and Lorenz [109, in 'The sad case of Lewis', Chapter 11] for projects with high turnover in the PM area.

Beyond the obvious, one would reasonably expect the subset of participants in the second case study who were directly or indirectly involved in the unsuccessful R' validation to be wary of expected pitfalls the second time around. Experienced participants might reasonably attempt to anticipate problems through close control of project management functions, tracking of contract line item deliverables, and frequent consultation with the certification authorities responsible for the project. Together with a shared awareness of unfamiliarity with the then-new DIACAP/ICD 503 C&A—or A&A—process for CDS, in the end, all of these things were observed. The contrast between the forced emphasis on process and standardised reporting[5] the second time around—and how that approach was qualitatively different from the *laissez-faire* project management attitude that characterised the earlier project—is readily apparent. Less so may be (1) observed differences between PM length-of-tenure across projects; (2) the observed correlation and postulated interaction between turnover of PMs and (in general) the effect of that on the relative balance between formal and informal (or implicit) intra-project communication channels; (3) the specific effects of PM changes on planned and implicit communication channels, or *vice versa*; (4) the difference between fact and belief in the existence and content of implicit

engineers, as previously noted, had the lowest turnover of all. Only one group—project managers, and to a lesser extent the Test Lead role at D—can be observed to have had a high or relatively high turnover rate.

[5] Reports are considered here as if they were communication channels, complete with latency, noise, and entropy; see Chapter 6.

and formal communication channels; (5) the difference between 'product knowledge' and 'process knowledge' and why this drives the emergence of informal channels; and (6) the rate of process improvement spin-off from successful and unsuccessful projects—a possibly useful new indicator of project health. Each of these these themes (Table 5.1) shall be examined in the sections that follow.

Theme	**Sections**
(1, 2) Formal channels are required by process, and they tend to resist disruption when the PM changes, but in the end these turn out to be less important than informal channels.	5.5
(1, 2) Informal channels are both necessary to success and fragile.	5.5.1
(1, 2, and 3) The successful project changed its PM far fewer times—but which way did causality run?	5.5.2
(4) Fact and belief are equally important to the sending and receiving of project status information.	5.5.3
(5) There is 'product' knowledge and there is 'process' knowledge—both are necessary.	5.5.3
(6) The successful R'' generated process improvement spin-offs, while the unsuccessful R' project did not.	5.5.4

Table 5.1: Broad themes observed in the R'' case study data.

5.3 The Form of the Data

Figure 5.1 illustrates the path that R'' took from R over time in relation to R' and R_0. The software development life-cycle used by D is a modified spiral with releases at (nominally) semi-annual intervals.

At the beginning of the R'' certification project, the researcher obtained access to classified meetings of representatives of D with the certification authority and penetration testers responsible for R''. At the time of the agreement, software development was complete and the developer's Factory

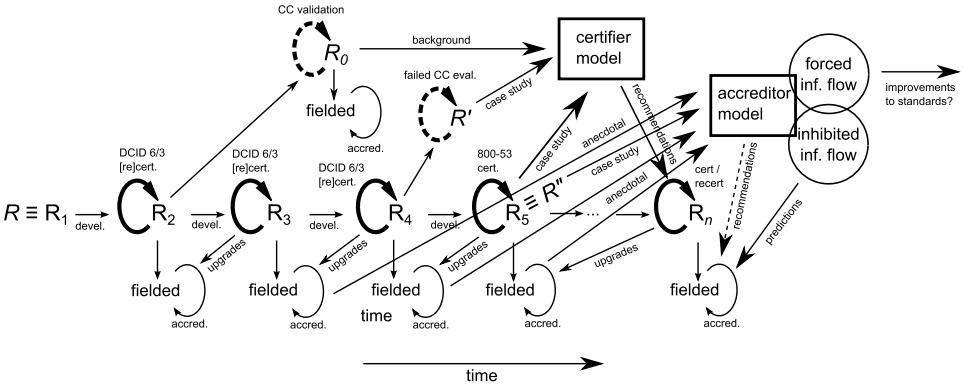


Figure 5.1: Ordering of events around the case study and the relationship among the R, R', R_0, and R'' systems and versions.

Acceptance Test (FAT) was beginning. Since the certifiers were in the Washington, D.C. region and the developer in Colorado, all meetings except for the kick-off took place on classified telephone conferences in the developer's SCIF. There was a classified e-mail system on the Contractor Wide Area Network (CWAN) network that could be reached from a terminal in the SCIF; this was how reports, agenda, and meeting minutes were exchanged. In return for minuting the phone conferences, the researcher (who already had the necessary clearance and access to the SCIF) was allowed to observe the meetings.

Because the meetings took place in a SCIF, minutes were typed up twice each time. The first time was on the CWAN terminal immediately after the meeting and produced classified minutes. The second time notes were typed up was outside the SCIF, within a few hours of the meeting, and yielded unclassified minutes of the same meeting. Only the unclassified minutes were used for this study. Generally they are similar, with the unclassified minutes omitting classified details but perhaps having the advantage of second thoughts, and including personal observations that would have been

inappropriate to include in the official, classified meeting minutes. The classified minutes were provided back to D's programme manager who forwarded them to the government programme office from which they were distributed to all other meeting participants. Feedback from participants reported that the classified minutes were accurate, detailed, and complete.

The unclassified notes comprise the data for the second case study. Because budgets, schedules, procedures, design documents, and reports were all classified, their only representation in the data is second or third hand, either if the researcher read the document or talked with someone else who did. Analysis of the data, using grounded theory methodology, began immediately and ran concurrently with data collection, as recommended by Glaser and Strauss. Hypotheses and interpretations were recorded in-line, although separated from observations by notation. What follows is an annotated time line of the R'' certification through the end of the CT&E phase of the DIACAP certification, derived directly from the unclassified, anonymised progress reports of the researcher. For each logical grouping of meetings (which may include the meeting itself, informal discussion before and/or after the telecon by participants on D's side of the call, recaps of the meetings given by D's managers to software engineers on the R programme who had not been in the meeting, and written reports), a synopsis of events is given, followed by a 'Analysis and Commentary' section containing interpretations. In the parlance of [99], discussions are analytical memoranda.

Where applicable, links are provided to the raw data in Appendix B in support of replication of the methodology.

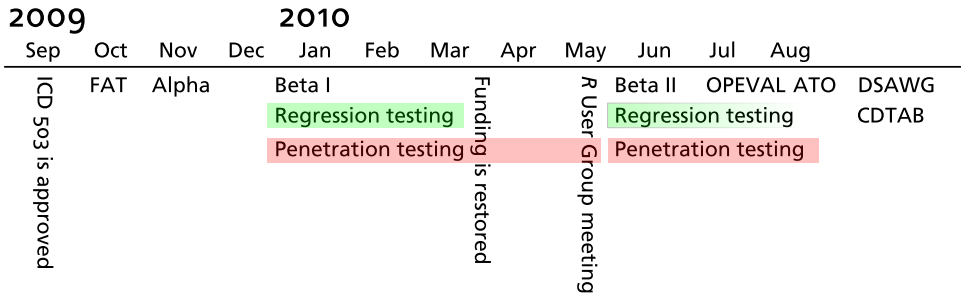


Figure 5.2: Graphical timeline of events in the CT&E of R''.

5.4 Annotated Time Line of Events

Under DIACAP, according to ICD 503, cross domain certification testing includes the initial operating capability at the first accredited site. Following the developer's Factory Acceptance Test (FAT), the entire certification testing of R'' (version 5.0) took approximately ten months (Figure 5.2).

5.4.1 Events in 2009

CT&E of R'' (version 5.0) commenced with alpha testing[6] in November 2009. D met with the certifier and Test Director for kick-off and concurrently began work on a response to the more than six hundred security controls from NIST Special Publication (SP) 800-53 that ICD 503 had identified as applicable to cross domain solutions. This was not something D was required to do; it was done, participant dev_1 said, in a spirit of co-operation, all parties involved having had no prior experience with CT&E of a CDS under DIACAP—the certification criteria having been approved only two months before. The Security Requirements Traceability Matrix (SRTM) for R'' comprised 653

[6]The three phases of DIACAP C&A are Alpha Test, Beta I Test, and Beta II Test, corresponding approximately with (1) IV&V of requirements satisfaction, (2) CT&E of the product, and (3) ST&E of the first operational site.

lines, compared with approximately 40 lines in the DCID 6/3 SRTM used earlier for R. As will be discussed in Chapter 6, Alpha testing duplicates FAT to the extent of using the developer's own test procedures, but in a witnessed environment, to verify that the implementation satisfies the requirements. The result of a month's work was the R'' Alpha Test Report, submitted to the government programme office in early December.[7]

For the first two days, the researcher participated in the Independent Verification and Validation (IV&V) Test Director's briefing and took notes in the test lab during alpha test Alpha testing. It is worth noting some interconnections between the IV&V contractors and the certification authority that were found in the data (Figure 5.3); a few individuals, *e.g.*, ivv_2, worked for both organisations and performed different roles at different times, although that fact was not directly visible to D—it showed up only later in a network diagram generated by ATLAS.ti. The internal structure of the certification authorities' organisation similarly is obscure—their actual organisation chart is classified—but a reconstruction based on numerous sources of information (attendance in meetings, unguarded comments, patterns of delegation, email addressee lists, and rumours) is shown in Figure 5.4.

Alpha test is the second of four phases of testing; the previous phase was Factory Acceptance Test (FAT), performed by the developer; following Alpha would be Beta I in the field at a government-owned site. The final phase, designated Beta II, comprised a professional penetration test performed by N'. Therefore, in all, four non-overlapping sets of people would have tested the software before it was certified: the developer, the IV&V contractor, the intended users of the system at Beta I site, and N'.

[7] Compare this level of caution to the subsequent (R''') certification testing (§5.4.6) where the certifier decided to accept D's internal regression testing—that was witnessed by a member of the certifier's staff—as sufficient evidence to pass the Alpha testing gate.

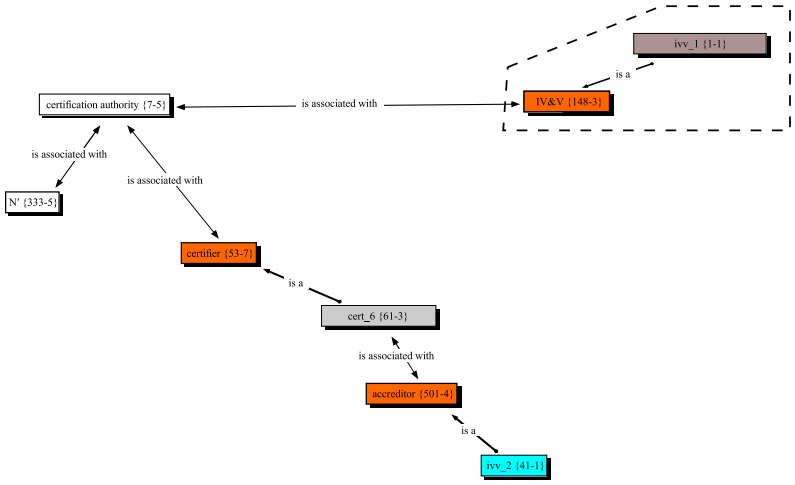


Figure 5.3: In the Independent Verification and Validation (IV&V) organisation, only participant ivv_1 was exclusively an IV&V person (dotted boundary); generally, however, IV&V worked for the certifier, not the developer or the Programme Management Office (PMO). In the figures that follow, values following code names give the number of quotations associated with each code and the number of links to other codes.

Participant dev_1 observed at the time that the *ad hoc* testing that had been explicitly written into the Alpha test schedule by the Test Director:

> *'. . . [would] be some of the most valuable. Test procedures [only] test that requirements are satisfied'.*

It was generally agreed that *ad hoc* testing was likely to generate Change Request (CR)-worthy findings and so should be moved up in the schedule so that CR builds—the software was currently at build 5.0z—could have time to happen. Only two of the SP 800-53 security controls were thought likely to fail, in both cases because R'' did not have those features, and D anticipated that aside from those two, no other Category I (called 'Cat I') or Cat II findings were expected. At the end of the Alpha testing week, the Test Director formally reported:

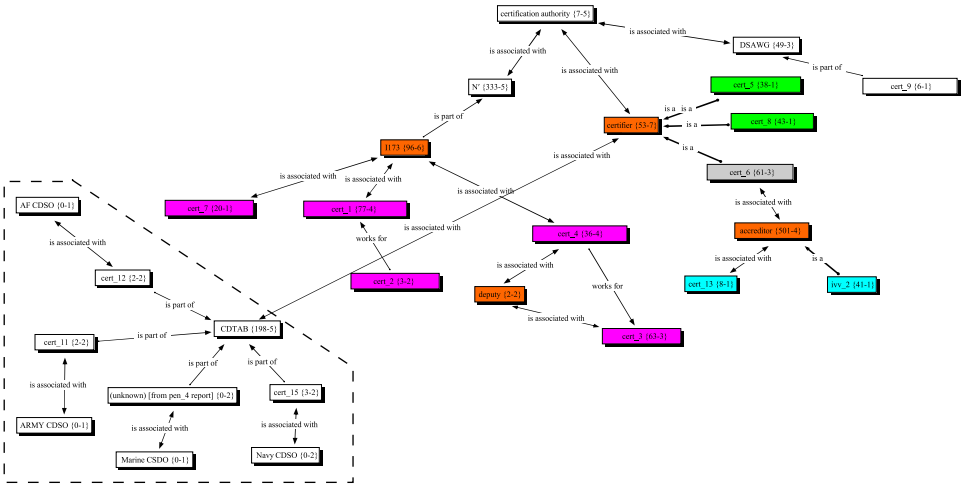


Figure 5.4: Relationships amongst participants and organisations making up the certification authority: I173 is the technical arm of N' advising the certifier; accreditors work directly for the certification authority but penetration testers and IV&V are separated and form cohesive groups of their own (Figures 5.7 and 5.3). The dotted line boundary at left encloses members of the Cross Domain Technical Architecture Board (CDTAB) and the Community Jury; because R'' was a widely anticipated CDS, representatives from the Army, Navy, Air Force, and Marines showed up frequently in telecons to look in on their interests.

> *'Between the testing done in FAT and the testing done by IV&V during Alpha, [R''] meets the security controls of 800-53 that have been assigned to it by UCDMO and it is ready for Beta I.'*

But as one of the longest-tenured and most experienced participants (dev_2) had put it earlier, during the Alpha Test kick-off meeting:

> *'Beta always finds something.'*

Analysis and Commentary

For measuring security improvement during the CT&E phase, a convenient measure is the number-of-findings-during-subsequent-testing-phases. The nature of cross domain solutions is such that—by definition—they are in-

stalled in multiple security domains, each controlled by accreditors who do not necessarily trust one another. Hence, the software tends to be tested and retested again using similar criteria but by different groups, *e.g.*, FAT, IV&V, and CT&E (Beta I); followed by ST&E (Beta II) at multiple sites in the same security domain, followed inevitably thereafter (forced by the software upgrade life cycle) by new rounds of CT&E and ST&E in other security domains and with new certifiers. Nevertheless, testing criteria are often the same or similar, since the penetration testers of N', the standards writers at NIST, and similar government agencies are the relevant authorities on the subject of computer security, and historically, different military standards tend to trace back to the same set of recommendations. Besides being a scalar, the aforementioned metric has the additional advantage of being available; it is usually possible to get aggregate counts of Cat I, II, III, and IV findings released by the certifier and the developer, if not identifying the system or version or patch level or application or site, so they can be used in an unclassified report.

An interesting question based on that metric would be, 'is there a difference in the post-CT&E software defect rate (as measured by the number of findings of Category I, II, III, and IV) between the same or newer versions of a given system in subsequent rounds of CT&E by different DAAs?' The first outcome—more findings over time—is made more probable by the continual improvement of testing tools, development of new attacks, adoption of new (presumably better) certification criteria, and by having more eyes looking at the subject. An example of this actually occurred in the previous case study: the system had been evaluated many times by one agency, but when a different agency looked at it, they found a weakness that had been overlooked for years. The second possible outcome—no change, or fewer findings

over time—is arguably what ought to occur if secure software development practices are followed. In order to settle the question, one thing that needs to be decided is what is meant by 'no change', and what are the implications of that decision; does it refer to the absolute number of findings, which is a multidimensional quantity, or does it measure the rate of occurrence, or a trend in that number, or in the rate?

Monetary cost is another metric, but one that is difficult to gather data for; it has not been clearly broken out in budgets, and project managers are loathe to disclose.

5.4.2 Events in 2010

Following verification of the developer's test results at D's facility in Colorado, testing moved to the U.S. Navy's Space and Naval Warfare Systems Command systems centre in Charleston, South Carolina[8], who were contracted by N' to perform regression testing for Q's certification project on a special build of R'' that had all options enabled. At the same time, penetration testing would begin at N''s headquarters, conducted by professional government security testers. Beta I was expected to take approximately three months to complete.

The testers in Charleston were civilian—not government—employees of the same contracting company that normally performed IV&V for P_1. The test procedures used were identical to the developer's test procedures; see Chapter 6 for more about this point. It will be seen later that apparent 'class differences' as well as indications of differing perceptions—between the government security testers and IV&V contractors (who performed the regression testing in Charleston) and D's developers—existed to the extent

[8]SPAWARSYSCEN Atlantic, Gate 4, 1060 Remount Road, North Charleston, South Carolina 29406, USA.

of two or three levels. Mirroring concerns in the previous phase, several participants said in mid January that:

> *'. . . it would be a miracle if pen testing were actually complete by [pmo_1]'s deadline in April'.*

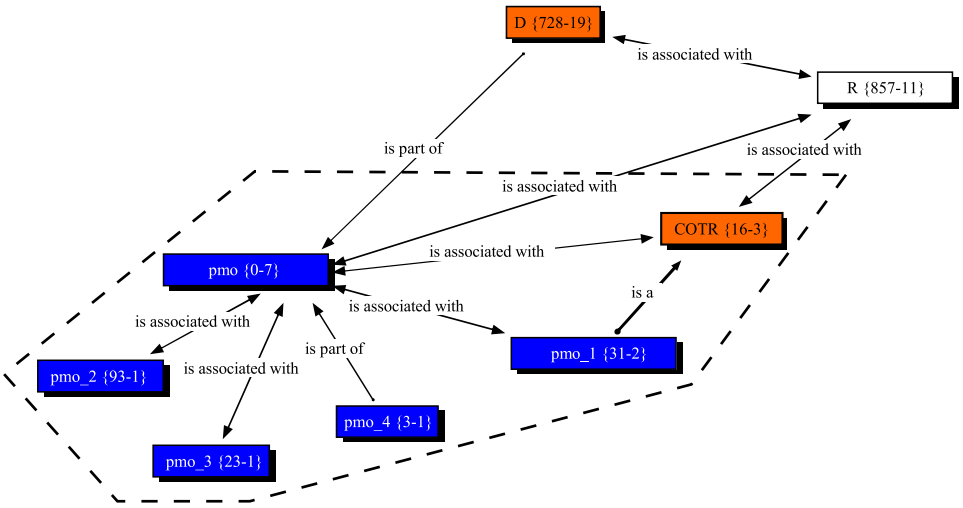


Figure 5.5: The Project Management Office controlled funding for the development and maintenance of R, but was able to exert influence directly only on D, as indicated by the dotted line boundary. As will be seen more clearly in Figure 5.10, PMO had no other connections outside D.

The Programme Management Office (PMO) oversaw but otherwise did not drive the C&A process (Figure 5.5). Note the way PMO had influence *only* over the developer (and the set of features in the product), but could exert no influence at all, directly, over IV&V, the certifier, CDTAB, or N'. Anecdotally, this may have come as a surprise to many of the developer's personnel, whose only regular contact with the U.S. government was the PMO, and who perceived the PMO as the driver of the CT&E process, as reflected in the above quotation. D's organisation was compact (Figure 5.6) and interfaced with PMO only through the Programme Manager, K_3 or dev_7, and indirectly through the software, R. The most visible source of control, of course, was money.

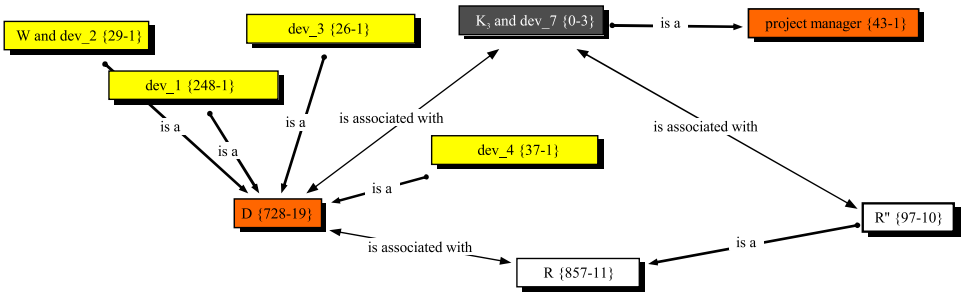


Figure 5.6: The developer consisted of a small core of software engineers (dev_1–4) with uniformly long tenure on-programme who did the majority of the design and development of *R*. The only direct interface to the PMO was through dev_7, the Programme Manager, who was on the staff of *D*, not PMO.

Funding issues were already beginning to affect progress. Anecdotally, in early March, the government programme office for *R* hadn't paid its bills to *D* since the end of the previous Fiscal Year (FY).[9] The programme manager artfully shielded line workers on the programme, who either did not understand or did not care what was going on. Less-experienced people on the programme were worried; old hands had seen it happen before, and were slightly worried, but they had been through 'stop-work' orders before.[10]

[9]The U.S. government fiscal year runs from 1st October through 30th September of the following calendar year.

[10]A possibly unrelated series of events happening on the programme, but included here because it illustrates the main driver (*i.e.*, funding) relates to the DoD 8570.01 certification requirement. DOD 8570.01 was a policy that required all persons (be they government, military, or contractor) with privileged access to defence computer systems (*i.e.*, having `root` access) must hold one of a short list of commercial security certifications, CISSP being preferred. *D* seemed to be dragging its feet on this requirement; the government itself reported that their own compliance rate was somewhat behind schedule—and had recently postponed the deadline for full compliance from end of Fiscal Year (FY) 2010 to end of Calendar Year (CY) 2010—but *D* did not appear to be doing a sufficient amount to achieve full compliance with 8570.1 by the end of December 2010. There were an estimated 40–60 people on programme who needed certifications, only ten or so of whom had it at the time, and the cost to certify that many people approached US$100,000.00. The programme had recently experienced funding difficulties; but worse, imagine if the Contracting Officer's Technical Representative (COTR) should write 8570.1 compliance into the contract at contract renegotiation time. The department might have suddenly found itself in the position of needing to get all fifty of those people certified before the

The government programme office, reportedly, routinely did not receive previously allocated funds until the last minute of the last day they were due. According to some of the participants, the government's funding office[11] often refused to talk to the PMO; after a while, the Programme Manager [paraphrased]

> *. . . could not get his calls returned until* [one of the executives of *D*] *met with the Captain in charge.*

Analysis and Commentary

A series of conversations were conducted with people in the developer's organisation regarding the certification progress of R''. There were early indications at this point of divisions within the certification authority. By the beginning of 2010, it was clear there were two people in the certification agency working at cross purposes; one (cert_3) was blocking it on an unspecified technical issue; another (cert_14) was championing R'' and advocated from within for its approval. The researcher continued listening in on weekly telecons and conducting informal interviews, watching the certification process without participating in it. The programme manager, systems engineering lead, and IV&V lead had been the best sources so far, in addition to two previous project managers. (Turnover of project managers, recall, was significant in the failure of R' to obtain certification.) The researcher began to map the roles that various people were taking, in contrast to the roles that they ostensibly were supposed to be doing.

In late March, funding was restored and was said to be assured through

end of CY 2010 and only three months to do it. Funding problems directly affected the progress of the R'' certification effort; by mid-March it was clear that they would miss the 15th April target date.

[11] The [Department of the Navy] office which funded the programme is different from the Programme Management Office (PMO).

end of FY 2010.

5.4.3 Early April

DIACAP certification of R'' was proceeding normally with Beta I tests occurring in Washington, D.C. during the month of April. N' ran two labs concurrently: a penetration testing lab at Ft Meade, Maryland (Figure 5.7) and an outsourced IV&V lab in Charleston, South Carolina. CT&E proceeded exactly as described in [142, §I.B].

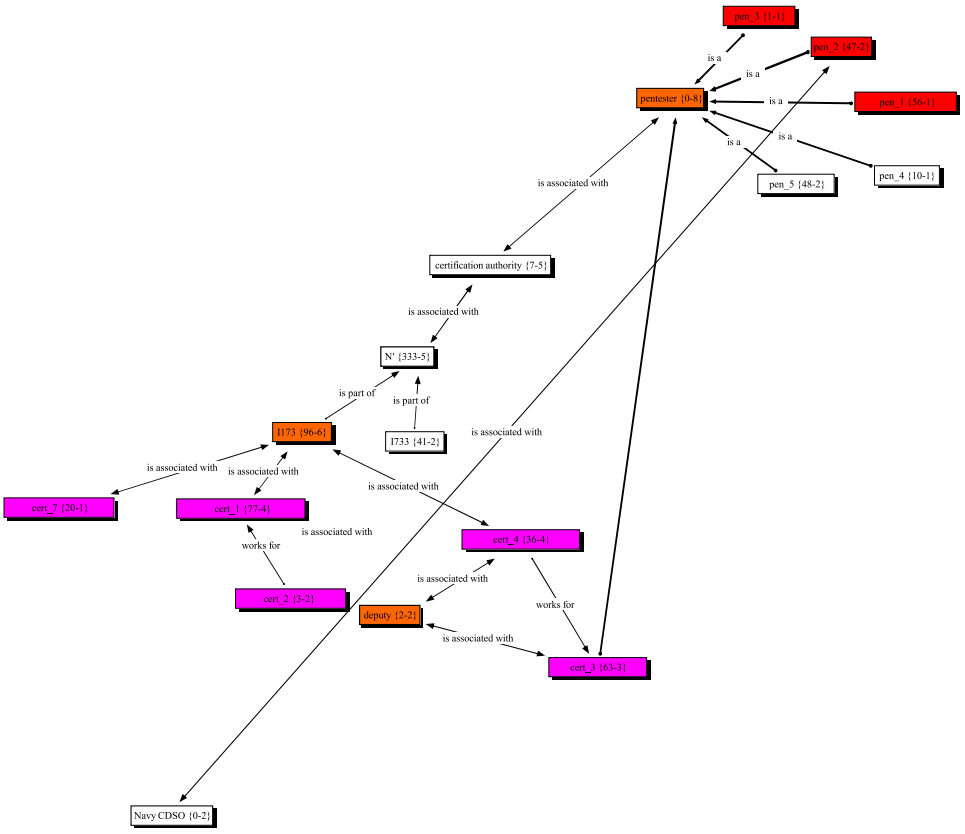


Figure 5.7: N''s penetration testers nominally worked for I733 but communicated primarily with I173 and referred to each other as if there were two-and-a-half distinct groups: one communicating exclusively with and through cert_3, another that was apparently under the charge of cert_4, and one individual seconded from the Navy CDSO, acting as lookout for that agency's interests with regard to certain [classified] tests.

Besides funding, another source of friction was differences between the software configurations provided to Charleston for functional regression testing *vs* the penetration testers for security testing; it was D's contention that penetration testers should work from a realistically configured and locked down system reflecting operational systems, while the regression testers—who were essentially re-running D's FAT—required a test-lab configuration with all options enabled in order to verify functional requirements.

The impasse was eventually resolved—after D maintained that the penetration testers had never asked for an open system—by three senior developers from D who flew out to the coast to reconfigure the pen testers' system identically to Charleston's. Refinements during testing like this one were unsurprising in light of the unfamiliarity of certifier and developer alike with the new DIACAP and ICD 503 C&A requirements for CDS; the test article had never been sufficiently specified in the Memorandum of Agreement (MoA).

The IV&V contractors in Charleston, then, were effectively re-running D's own FAT. They failed many test procedures—all minor to moderate, characterised by dev_1 as 'nothing to worry about'—because the regression test contractors kept attempting to test features of the software that were not installed. Both test sites expressed frustration that they couldn't test a fully configured system; D responded that N' had never asked for a fully configured system in the first place. The developer sent another person to D.C. to configure the Ft Meade system to be identical to that in Charleston.

Near the end of the month, the IV&V team in Charleston, South Carolina were wrapping up and expected to be finished with their testing soon. They would then write their report, due 30th April. Penetration testing at Ft Meade was still ongoing; those people would not stop until ordered to write a

report. Unlike the IV&V group, who have a finite number of test procedures to go through, pen testing is completely open-ended and does not stop until the money runs out.[12]

The IV&V report would come to the developer and the certifier at the same time. The developer would spend a couple of days reading and understanding the report, then probably need to develop a patch for any Cat I or Cat II findings from CT&E. Issuance of the patch would be followed by commencement of Beta II testing in an operational environment.

The plan called for Beta II testing to be done at Joint Forces Command (JFCOM), but JFCOM had a time constraint; it was necessary that Beta II be complete by 1st June because the facility was needed to support an exercise at that time. There was not enough time between 30th April and 1st June for D to receive the IV&V report, develop a patch, get it into testing at JFCOM, and finish by the first of June. So the developer and the programme office formulated two alternative plans: either wait until August when JFCOM will become available again, or move the Beta II testing to Strategic Command (STRATCOM) instead.

Analysis and Commentary

Neither of these backup plans was particularly desirable, but the researcher predicted at the time that they would choose not to wait. Waiting until August for Beta II (Operational testing) would delay certification, and that would impact the plans of multiple customers counting on getting certified R''

[12] Penetration testing is similar to Covert Channel Analysis (CCA) in that way. This is a government peculiarity, though; in contrast with government-sponsored pen testing, which *limits the duration but recognises the utility*, commercial firms, according to a security assurance tester of the researcher's acquaintance, often see pen testing as a tick box standing between them and, say, PCI compliance; uninterested in quality pen testing, commercial firms will pay only for the minimum amount required and might not even read the tester's report [270].

systems in the summer of 2010. The researcher predicted that the programme office would do two things: firstly, they would try to squeeze Beta II into JFCOM before the deadline; they would waste some time trying to do so, and this would be followed by the certifier telling them to desist. Following that chastisement, they would move Beta II testing to STRATCOM to avoid delaying the certification.

The developer expected at the time that no major findings would result from Beta II. There was always the possibility of a patch, which would necessitate re-running regression and IV&V tests, but that was considered unlikely. After Beta II finished, and the operational site submitted their report to the certifier, then it would be only a short time before a certification letter would be issued. UCDMO would announce version 3.3 of the list of approved guards, and installations at operational sites would be able to proceed as scheduled.

The above series of events is interesting because it illustrates how the certification process actually works in practice.

5.4.4 Late April

Beta I continued to proceed normally; IV&V was expected to submit their final report on 30th April. Rumour had it that there were no major findings. The developer was only working on 6–8 CRs (software Change Requests) at the time, none of them large; things looked good for version 5.0zc to become the build candidate to go into Beta II regression testing—due to complete 18th June (for FAT) and 8th July (for the government's regression test).

Analysis and Commentary

As predicted, the developer finally decided to slip the date for Beta II by six weeks and test at STRATCOM instead of JFCOM. The project manager noted in a meeting that the artificial sense of urgency that prevailed whilst the developer were still trying to make the JFCOM deadline was 'probably the only reason why [they]'ll actually receive the IV&V report' by 30th April; without that pressure, [the IV&V contractors] 'probably would not have worked as hard'. The slippage of 6–8 weeks from the original date was not unacceptable, although funding was very tight and the developer remained 'at risk' through 30th April.

> *'There could be nuclear missiles incoming to Washington, D.C.,'* [the Project Manager said], *'and NMSO* [the funding agency] *would never move any faster.* [The government sponsor] *has been incrementally funding this project to death.'*

The cost of Beta I testing, at the end of the phase, was called excessive by the PMO and programme manager.

5.4.5 May

The researcher's attendance at the *R* Users' Group meeting this month yielded insight into the C&A process from the perspective of CDS users. Several dozen *R* users representing multiple service branches and the IC attended a week of presentations at *D*'s facility between Beta I and II. It was announced that the developer expected to receive the Beta I test report from IV&V soon; users in turn emphasised the extreme need for R'' to be certified and deployed as soon as possible. The developer continued to be severely underfunded, although it was not entirely clear whether this was the fault of the programme office or another agency further upstream. The

COTR was sitting next to D's programme manager when the latter described the excruciatingly tight funding situation in front of an audience of users, but even after watching that exchange, the exact nature of the COTR–PM relationship remained unclear.

The certifiers took advantage of the gathering to present their own side. I173 had a new model for cross domain CT&E called the Risk Decision Authority Criteria (RDAC).[13] It assigned a scale of five rating levels to three types of risk: Technical Risk (TR), Data Risk (DR), and Attack Risk (AR). For example, DR might be assessed for risk of spillage or policy bypass threat as one of Low, Medium, High, Extreme, or Unlimited. For each combination there was a matrix specifying which risk mitigations would thereby be required.

Beta I was now complete, announced in a conference call with the Programme Office's IV&V test lead, N''s departments I173 and I733, and D. Overall, three formal testing reports would be delivered: I173's, I733's, and DNI CAT's; final versions of the I173 and I733 reports would be issued 14th May. The developer was currently writing a formal response to all findings; details of findings were classified. The schedule was unchanged, with no alteration expected to the 20th August final certification date.

Analysis and Commentary

The penetration testers expressed considerable annoyance with two particular data flows that the developer provided for CT&E testing but which were not fully vetted. The testers said they 'really had to rush' to get everything tested, following which D responded to the testers' final report with,

[13] The group called 'I173', together with a different group confusingly designated 'I733' (in charge of penetration testing), was a department or sub-organisation of N' tasked with certification oversight of the Charleston regression testers; the designations changed constantly, and no unclassified organisational chart for N' exists.

'Oh, those two flows were fake, not real ones.'

The testers had expended a significant amount of time on those two flows, but now the results of those tests were disregarded because the developer claimed that the flows had never been fully vetted. The developer countered that [the functionality] being exercised by [those flows] were an important new feature and there were no accredited examples of it yet, so it was better that the functionality be tested than not be tested. The interpersonal dynamic observed during this telecon—including not just the software developers but the programme office, penetration testers, IV&V contractors, and certifier—is important in light of some themes identified in the thesis: the relative importance of informal channels over formal ones, especially with respect to speed; observed differences between fact and belief (at least in the case of the developer) relating to the government security testers; and the importance of a strong project manager.

There was a complex multi-way disagreement over some issues, expected to be resolved a few days later after Charleston's official report was received. The installation and testing schedule for Beta II would slip one week to the right. It was not known whether that would change the estimated date (20th August 2010) when certification and approval to field the new system were expected to occur.

A later R'' telecon supplied a little more insight into what the disagreement between the two camps had been all about. The purpose of the meeting was to go over the developer's final response to the certifier's reply to the developer's response to the certifier's findings (Figure 5.8), but what really came out was a philosophical difference over the meaning and scope of CT&E in relation to ST&E. There were voting members of the Cross Domain Technical Architecture Board (CDTAB) on both sides of the issue. The

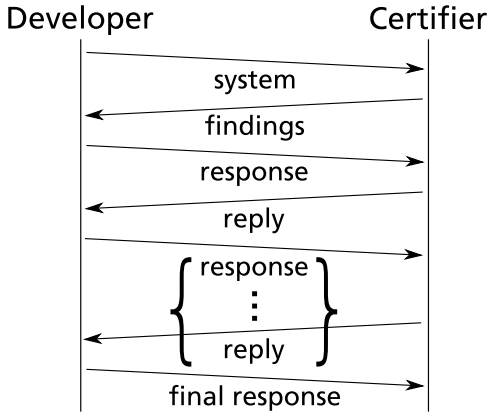


Figure 5.8: Dialogue between developer and certifier over findings.

certifier wanted to remove certain functionality from the CDMO baseline configuration because the certifier felt that it had null functionality. *D* and the Programme Office argued that function was inherent in the framework, that the framework was there for each accreditor to specify how it would be used at a particular site, and that the choice of what to hang on the framework was and ought to be the accreditor's. The certifier was afraid that if the CDMO baseline contained an adaptable framework, 'some sites will configure it in a stupid way', and the result would be insecure. The developer and the Programme Office countered that because the accreditor formally accepts responsibility for the residual risk and correct operation of a system at a site, no accreditor would allow the framework to be configured that way.

IC (Intelligence Community) certifiers[14] were ready to proceed but SABI (Secret and Below Interoperability) certifiers were not.

> *'That completely defeats the purpose of the UCDMO'*, exclaimed one participant.[15]

[14] That is, Top Secret and Below Interoperability (TSABI).

[15] *You can argue that the UCDMO is the U.S. government's formal organizational*

Which brings up an interesting point: the way things work in practice has never been well established. Several government standards exist that prescribe certain steps that must be followed, but the very nature of a CDS necessarily spans boundaries, and as a result multiple standards almost always obtain, partly overlapping in their scope of authority but with substantial areas of sole authority on the edges. It is an open question: how do accreditors resolve this now? Do they work serially, or concurrently?

The topic of inter-accreditor communication was not one initially thought important, but now we can see that it is. What connections exist, how much and what kinds of information are exchanged, and does a path exist through the network that could make the process more efficient? These questions are addressed in Chapter 6.

At the end of May, the schedule had not changed. Beta II installation was expected to occur as planned; *D* and the programme office were finalising their formal response to the list of Beta I findings from SSC Charleston and I733. The Plan of Actions and Milestones (POA&M), a required document in the NIST SP 800-53 process, was being drafted.

5.4.6 Early June

Regression testing of Change Requests (CRs) in response to Beta I CT&E findings had been under way for a week. All of the IV&V contractors were at *D*'s facility; they worked all weekend and some of them (*e.g.*, the Test Director) had worked all night. Regression testing was not going well.

response to the tragedy of 9/11, in the words of Ed Hammersla [76]. The reason for the stark difference in security evaluation rigour between TSABI and SABI is often misunderstood—it was a mystery to *D*. The rationale is straightforward, if obscure; as explained by Steve Welke [in paraphrase]: *in the TSABI world, the low side is a SECRET network, but in the SABI world, the low side is the internet. Risk is perceived to come from the low side, and the level of risk is thought to be proportional to the number of levels difference across the CDS* [260]. (Welke's lecture is the only place we have ever seen the reason for the difference in rigour between SABI and TSABI explained.)

Two major new features were being tested; one was not working. One of the features was added in response to a 'high' importance finding during Beta I CT&E; that feature was working. The other major new feature had been added by the developer two months previously (at the direction of the PMO); it was the latter feature that was not working. The developer lacked the option of dropping the second new feature because it replaced old functionality that had been removed two months ago. *These problems are interesting because they show the effect of late software changes at an extremely time-critical point during CT&E. Less important changes would be backed out and not allowed possibly to affect the certification schedule.*

Next week the developer would be required to ship equipment to the Beta II operational site; this would be followed by three weeks of government regression testing. To save time, the developer was having IV&V witness some of the regression testing at *D*'s facility that week—with the agreement of the certifier that it would allow a portion of the government regression testing to be skipped in Charleston, as it would have already been performed and witnessed by IV&V. Three weeks of government regression testing was to be followed by ST&E at STRATCOM, comprising Beta II. One of *D*'s senior software engineers admitted that:

> *'STRATCOM [are] bending over backwards to help.'*

The current phase would be followed by penetration testing at a live site, not yet named. The developer expected to receive approval-to-field on 20th August; that schedule had not slipped.

Analysis and Commentary

Was enough slack time designed into the schedule at the beginning? The schedule is classified information, but one piece of observable evidence is

suggestive: the fact that the CT&E process had stuck precisely to the schedule with almost no slips, despite extremely tight funding at times.[16]

5.4.7 Mid-June

During a lull in the CT&E activity, the researcher met with the Test Director (ivv_2) for a conversation about some theories in the early stages of formation. The topic of the conversation was multi-lateral accreditations, with more than one accreditor.

Sometimes, explained the Test Director, the MoA will specify which DAA is in charge. Other times, especially if at least one interface of a CDS is connected to SCI, the DAA on that side may announce that he or she is taking responsibility. Other times, a CDMO or the UCDMO may decide. On rare occasions, the decision goes all the way up to the PAA, *i.e.*, the Director of National Intelligence (DNI).[17]

The Test Director was willing to discuss—off the record—the politics witnessed on telecons, some of the personalities involved, and various reasons why they might be doing some of the things they were doing. When shown an early version of the accreditor model that appears in Chapter 6, the Test Director, an experienced DAA, expressed great interest in seeing an improved method (derived from the model) adopted, and urged this research to be pursued for the next five years.

The Test Director validated that the model of multiple DAA interactions in cross domain CT&E is correct, saying that it fairly reflects the way real accreditations have been structured in experience. The Test Director wanted to talk about DAA–DAA interactions where accreditor A has information

[16]The truth of it is supported by some off-the-record comments from the penetration testers; because they were given off-the-record, the comments unfortunately are not included in the data of Appendix B.

[17]For international accreditations, this gets more complicated; see Chapter 6.

but will not share it with accreditor B because of a turf war. This is slightly different from the theoretical scenario envisioned in the following chapter, but does closely mirror observed interactions during the R'' CT&E, and it is believed that the concept of 'turf war' can be figured in a way not unlike how security clearance is presently handled.

The researcher despaired of ever achieving a complete understanding of the politics going on in that room. It is possible to describe what happens, to make hypotheses about patterns observed across two case studies, and to test hypotheses on the abstract model. But the Test Director had one more thing to say:

> *'If the accreditor model* [in the following chapter] *can make predictions about the behavio[u]r of future ST&E efforts,'* [said the Test Director] *'then something worthwhile will have been achieved.'*

5.4.8 Late June

The second telecon of the week included cert_6, cert_1, dev_1, pmo_1, pmo_3, dev_3, and K_3; cert_1 and cert_4 from I173, DNI CAT[18], dev_4, and the Test Director.[19] The meeting began with discussion of a revised format for the POA&M, but then:

> *'The developer'* [said cert_1] *'is recommending that in some cases a feature not be used, and to document that, and call it a "mitigation".'*

Participant cert_1 said that some of these mitigations were unacceptable, and further wanted to give guidance to the PMO on mitigation; pen_1 and pen_2 would assign priority numbers to all of the items.

[18]The name of the Director of National Intelligence (DNI)'s Compliance Assessment Team (DNI CAT) is not anonymised because C&A for CDS is their primary function.

[19]The weekly (or sometimes twice-weekly) telephone conference call ('telecon') was referred to by participants as a 'hotwash' although the etymology of that term is unclear.

The time for discussing mitigations was getting short; with the Test Lead at the Beta site and dev_1 at N' supporting the pen testers, pmo_2 commented on the six hundred or so NIST SP 800-53 security controls, saying:

> *'We think it is maybe a burden on the sites to answer a couple of hundred questions. It is a burden on the operational sites. The data are there, it is just that gathering it is a burden. Because part of this CT&E effort is to assess how 800-53 works in the CDS accreditation process, and because the CDMO is the face to the sites, they come back to the CDMO with their complaints, not to the CTSG or the UCDMO.'*

The Test Director promised to have a spreadsheet of NIST 800-53 security controls converted to an automated tick-list for the sites by the end of the month. During the week of the twenty-first, pmo_1 was hopeful of sitting down with the pen testers and showing them things in their own lab. This was hoped to defuse some of the risk mitigation disagreements.

Analysis and Commentary

Certification testing of R'' was proceeding well.

The situation with new functionality causing problems in regression testing was resolved the following week. There were 101 findings during Beta I CT&E, 25 of which were fixed with software changes. Of the remaining 76, approximately 20 were fixed with configuration changes, 20 were addressed with documentation changes, some were assessed as invalid, and a few would not be fixed. There were 22 CRs in [this version], some of which fixed multiple issues. It was understood by all involved that [certain long-standing issues] would never be changed because the Navy would never come up with the money. Every time a new version of R undergoes CT&E, the same issues are raised, and the answer is always the same:

> *'. . . no money to fix, or they were design decisions made by an earlier sponsor and consequently not the current PMO's fault'.*

The project manager (from a separate conversation) appeared to be satisfied with the progress of R'' through CT&E. There were more actual customer configurations—ten or twelve, this time—in testing than had ever been in previous CT&Es, [the project manager] said.

In a follow-up meeting, pmo_2 asked for concurrence from N' that there were no show-stoppers. Participants cert_7 and cert_4, collectively representing I173, agreed, with cert_7 further commenting that [this version] was not a particularly high risk. One of the more experienced programme office people (pmo_2) said,

> *'. . . obviously, if you find something in the next four weeks, then all bets are off. . . '*

but in general, all of the participants on that particular call struck the researcher as mellow and agreed that all findings of risk were being mitigated satisfactorily. The head pentester, cert_3, was not on the call, however, and had not been the last few times; cert_4 seemed to be trusted more and more by cert_3 as [his or her] deputy, as seen in Figure 5.9. In the researcher's experience on-project, it was impressive to see pmo_2 get I173 to agree verbally that the mitigations were sufficient and no show-stoppers were expected. Further, DNI CAT did not contradict that statement, which was truly amazing.

Any remaining questions of the certifiers regarding implementation details of the developer's proposed risk mitigations would be addressed by means of a combination of live demonstration, inspection of the implementation, and consultation between the software writers (who would be on site) and the certifiers with equipment to hand. PMO expressed hope that that

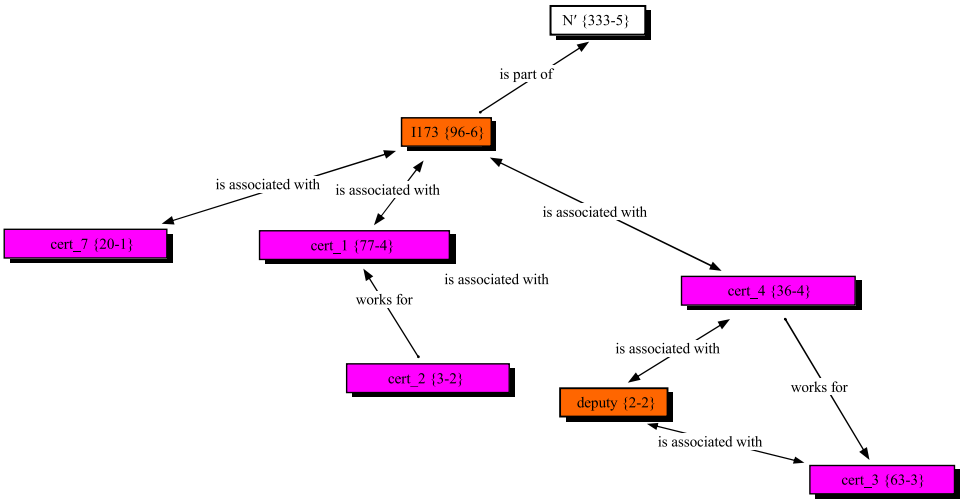


Figure 5.9: Over the course of the CT&E, it became evident to D that cert_4 was taking over some responsibilities from cert_3.

environment would be the best route to complete understanding.

Regression testing (Beta I CT&E) of R″ at D's plant in Colorado was now complete.

5.4.9 July

STRATCOM Operational Evaluation (OPEVAL) was scheduled to begin in two weeks, following ten working days of government regression testing.

Evidence of Personality Conflicts

One of the developers related to the researcher privately that many of the findings out of Beta I were 'political in nature'. Interestingly, the developer strongly preferred the personality of the professional government security testers at N' over that of the IV&V contractors in Charleston, as the following (paraphrased) story illustrates:

> Charleston, it was said, raises a fuss every time they find the slightest deviation; for example, some system administration tools were left on the test machine for the purpose of facilitating reconfiguration of network settings during testing. It was done that way on purpose for the convenience of the testers, but the testers in Charleston declared that they would write up every one of the files as a finding anyway (going by the book). In response, the developer's Test Director, test_1, resolved to make sure to lock down the Charleston machines especially well next time, just to make the Charleston testers' life difficult. The testers got precisely what they asked for: their life made difficult.

From the other side, cert_4 told the researcher that R was the only CDS that consistently met all of their deadlines:

> *'If the R developer says they will deliver something by a certain date, they do. None of the other guards does that.'*

Developer–Certifier interactions, rather than DAA turf wars as previously thought, were shaping up to be the most interesting aspect of the R'' case study. Roles seemed unimportant now, but informal communication channels kept turning up to be at least as important, if not more, than planned communication.

Evidence of Sub-Roles Not Directly Visible to D

After a hiatus of two weeks, the R'' certification status telecon met again with dev_1, dev_4, pen_5 (moderating), user_1 from US STRATCOM Western Region, UCDMO, cert_4, pmo_1 (representing the PMO), cert_1 (representing DNI CAT), and ivv_2 (representing IV&V) in attendance.

The currently active pentesting head, cert_4, started off by reporting that

they were still doing regression testing, and nowhere near done. They found previous findings not fixed, and some new ones; one of the new problems was reproducible by two different means.

Next, cert_4 complained that the pen testers '. . . feel [they] are finding old things that have not been fixed, and new things besides'. It became clear that cert_4 wanted to talk it over with cert_3 first, who was more experienced. If the developer had been paying attention, this was the first insight into the hidden I733 organisational chart: cert_4 was apparently being groomed to be cert_3's deputy in matters of CDS vendor relations during CT&E.

Then, cert_4 brought up the most important point of the whole meeting:

> *'. . . how are we going to validate the fixes between now and STRATCOM?'* [cert_4 wanted to consult with cert_3 about it.] *'This is a new situation,'* [said cert_4] *'one that has never happened before, to have a guard with unfixed findings.'*

Participant pmo_1 again emphasised the importance of keeping to schedule, for reasons of funding and funds allocation.

Evidence of Friction Between Developer and Certifier

In contrast, cert_1, of the I173 organisation (in charge of the Charleston[20] group) was beginning to be seen by the developer as obstinate, if not obstreperous:

> *'I173 still [have] some issues with the developer's response to the list of findings.'*

Participant dev_1 explained that a few weeks prior, when developer representatives were on site during set-up of the labs for regression testing, *D* had sat down with all the testers and went through every finding together,

[20] These were regression testing contractors.

in front of the actual machine, to come to a common understanding. But now,

> *'. . . still, some of the developer's responses to findings as written in the POA&M are not acceptable.'*

Various participants amongst D personnel routinely expressed frustration with N' pentesters after the weekly telecons, once the remote end was safely off the phone. Particular complaints expressed included '[N'] arrogance', obsessiveness over [a classified technical issue that D felt was not exploitable] dating back many years, the assertion that '[cert_3] hates [R] and will never give it an even break', and the feeling that pentesters were given an unfair advantage by bring provided with `root` passwords from the beginning.

Analysis and Commentary

D's software developers believed that N' testers 'have a whole pocketful of special tricks' aimed at testing guards, but D was aware of the reasoning why N' would not give D their test procedures, because N' do not want CDS developers writing products around it. The developer felt, however, that the certifiers were changing the rules in the middle of the game. In the past, doing SABI and TSABI separately, D would have been long done with TSABI by now. Under the new unified process through the UCDMO, there was no distinction between SABI and TSABI any more—eliminating a treasured shortcut to the ability to sell a product earlier. TSABI has always been much less strict than SABI; in the past, SABI always took longer. But under the unified process, N' was in charge, and N' are extremely strict. According to one frustrated project manager:

> *'Everything is being held up by [N'].'*

5.4.10 Mid-July

Pentester cert_3 complimented the developer for being responsive, but expressed frustration that the pen testers had been unable to finish all the tests they want to run, because they kept running into breakages. Testing stopped while the developer fixed a problem, and then picked up again. The pen testers were especially frustrated that previously identified findings were being found not to have been fixed, and new findings were found besides. It was slowing down testing, they said. Paraphrased [in the words of cert_4]:

> *The pen testers have not been able to look at key functionality because of problems that keep cropping up. Look at the history: CT&E started back in March. Since CT&E began, the test team has never been able to get the entire system up and running. They spent a week trying to install, debugging, changing mags on the fly. Not to fault the developer—it was a very aggressive schedule. But every time the team starts trying to test, something else breaks. [D] has always been very responsive. But every time the testers try to test, it's back and forth, back and forth.*

The developer replied that in the TOE test scenario set up for the test labs to operate in, there were numerous factors that differed from the real world. In actual installations, a running configuration was very specific and tuned to do exactly what was needed. No matter how much the pen test team would like the system they test to reflect the real world, the exigencies of testing enforce an artificial environment. The developer strove to give testers the opportunity to perform significant and meaningful tests on the full scope of functionality of the system, but testing some features in combination was neither meaningful nor realistic.

Evidence of 'Sharp Remarks'

D stated at the end of one telecon that they wished to make the following points:

1. N' pen testers were frustrated that they could not perform testing non-stop without encountering issues. (This is where the original plan to have an R engineer on-site during the entire test event would have been very beneficial.)

2. The participant cert_3 '... used harsh words'. It is crucial to recognize, however, that the N' testers were looking at a completely unrealistic configuration:

   > *'Their test system is a Swiss Army knife. It has every bell and whistle turned on, things that would not ordinarily be used in combination. Operational installations of [R] are never configured that way.'*

3. The participant cert_6, however, was pushing strongly to stick to the original schedule:

   > *After many sharp remarks from cert_3, in the end [that person] seems to be OK with rating certain things as high risk and moving on as long as they are documented. [The developer] wants to ensure that only those specific items are rated poorly, and not the whole guard.*

5.4.11 Late July

The R'' certification test and evaluation process was moving towards a successful conclusion. The developer received drafts of the final N' penetration testing report and government regression testing report from Beta I CT&E, and there were no surprises. The developer had fixed all of the issues that

they chose to fix, and documented the others. There was a 5.01 patch[21] preliminary build meeting to approve files that would go into the POA&M patch; no date had been decided yet for when the patch would be issued, except that it would be after 20th August. The developer continued to express the feeling that one person (cert_3) at N' had a bias against R and would always find something wrong with it.

Beta II would officially begin the following week with ST&E at STRATCOM. 'The first few SABI installations [after STRATCOM]... will have the hardest time getting through' [the CDTAB and DSAWG[22] process]...', observed pmo_2; 'we shall have to look hard at the body of evidence for those first sites'. A different person from before, pen_1, would be doing a Test Readiness Review (TRR) this time for CDTAB in September. In preparation for that, [pen_1] was looking at the entire body of evidence created so far, both RDAC and RMF.

After some further discussion of risk assessment, pmo_1, pen_1, and cert_1 informally agreed that the first few SABI site installations could proceed with the first certified R'' software version in advance of the POA&M patch (that is, R'''), '... although possibly with a higher risk assessment'. Would it be possible for pmo_2 or another representative to attend the CDTAB?, asked the PMO. It was suggested that pen_1 should try to get an invitation to the CDTAB, although the CDTAB did not usually entertain visitors. A more experienced person (pmo_2) observed that IV&V was more likely to be invited, since they were totally independent of D. Ultimately, pen_1 said that because this was the first time for a new CT&E standard, it was possible that others might be able to get an invitation to CDTAB.

All of the old items on the findings list would be re-checked by ivv_2, to

[21] This is the post-CT&E software version, designated R'''.

[22] Defence Information Assurance Security Accreditation Working Group (DSAWG).

verify that no findings have been forgotten and that all software changes have been tested.

After the telecon, dev_4 noted:

> *'... this was the best meeting yet. It looks like it is actually going to happen.'*

5.4.12 August

CT&E of R'' was completed on schedule. The developer received a copy of the final report from N' I733 via USN SPAWARSYSCEN in Charleston. The researcher read the report, which covered U.S. government regression testing only; penetration testing by N' I173 would be reported separately.[23]. There were no surprises other than the particular way the report was written: the developer felt that the presentation of results was incomplete, resulting in an unbalanced impression to the reader. Every test that ever failed in regression was reported as FAIL, even things that were fixed during CT&E, re-tested, and later passed. The developer intended to ask for an updated report reflecting the final findings, believing that this would cut the immediately apparent number of failures down to a handful—some of which were actually 'the system is working as designed'—making the report easier to interpret. It is unlikely that any TOE will ever go before CDTAB with a completely green-light report; some of the security controls in 800-53 specified by a particular profile set are unlikely to be met *in toto*, if they do not actually conflict. The researcher spoke with several of the developer's engineers about the report, and predicted that there was nothing in it that would prevent the STRATCOM accreditor from approving the ST&E on 20th August, but that the report in its current state would give CDTAB and DSAWG pause.

[23]The reports are classified information.

It was likely the developer would get a certificate of some kind in August, but the first SABI sites were going to have a lot more work to do.

The next stage of C&A was ST&E, or accreditation. Because ST&E occurred repeatedly at *each* operational installation site, and because the sites were classified, and because the progress of testing at each site was under the sole purview of the DAA—not reported the same way as CT&E in weekly telecons—the researcher was unable to observe the first ST&E.

As of the beginning of August, 2010, ST&E was reported to be under way and going well. Some of the participants would sit down at the UCDMO conference the following week in Boston to discuss the progress of the accreditation. The rest of the developer's organisation waited for reports to trickle in.

5.4.13 Successful End of ST&E

R'' received its Approval to Operate (ATO) from the accreditor at STRATCOM in the third week of August, 2010, representing a successful completion of ST&E under DIACAP, and the first successful DIACAP accreditation of a CDS under the new NIST 800-53 rules instituted across the IC in September 2009. Penetration testing, which was still going on as of the ATO, was proceeding well—from the developer's perspective. They reported that the pen test team at STRATCOM were frustrated in their attempts to compromise the box under operational conditions.

ATO indicated that this CDS was in production now; certification, which means it appears on the UCDMO baseline list of approved cross domain solutions, would follow. It is unknown whether the developer ever was successful at getting a revised final report out of I173 that showed successful mitigation of all findings. The evidence package that would go before the

DSAWG and CDTAB would comprise that report amongst others.

5.4.14 Overview of Relationships

In Figure 5.10 it is possible to see the overall interrelationships between each of the different groups in both case studies.

Consider the situation from D's point of view. It can readily be seen that the R'' application is the focus of attention from twenty-four people in nine organisations, not counting D. By transitivity, R'' is associated with twelve other people in D and P_1 by virtue of being a version of R. It is apparent that participants divide approximately into four groups, designated 'Developer', 'Certifiers', 'PMO', and 'IV&V', which is logical in view of the functional and process differences between them. There is almost no overlap amongst the four groups.

One relationship is troubling, in retrospect. The connection between pen_2 at the upper right and the U.S. Navy Cross Domain Solutions Office (Navy CDSO) in the group at lower left is interesting because of the apparent conflict of interest between pen testers, who were working to advise certifiers N' and Q, and the Navy CDSO, a member of CDTAB, and therefore part of the certification authority. Corroborating the observation, the Navy CDMO exhibited (anecdotally) a persistent bias against R'' which shows up in the pen_4 report.

5.5 A Grounded Theory of Communication in the Case Studies

It is interesting to consider all the interpersonal and inter-organisational interactions we have just seen in light of the idea of communication channels.

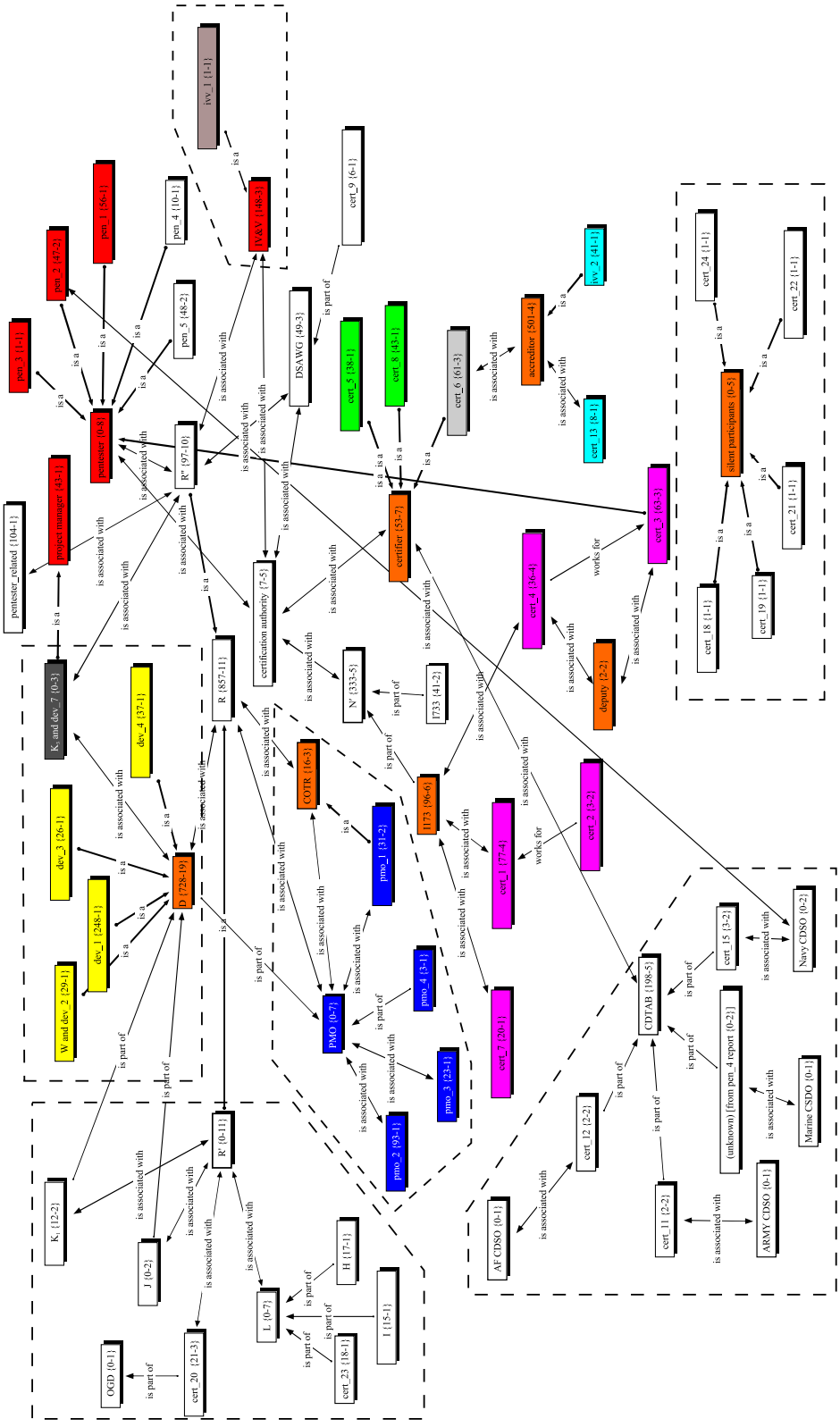


Figure 5.10: Overall view of the network of codes and quotations in the two case studies. The certifiers of R' were a completely separate group from R'', with no overlap. (The developer and PMO remained unchanged.) Irrelevant groups of participants, as determined by lack of connections, have been partitioned off.

Firstly, we ought to define the idea of channels and signals as applied to the CT&E activity. That activity is done by people, according to procedures that are defined in a process (in this case, DIACAP). Because there are several people (developers, certifiers, testers, accreditors, and users) involved, they must communicate. The process defines a number of required reports, and together with conference calls and a few 'official' emails, these comprise the *formal channels* through which all information exchanged amongst the participants is expected to flow. Formal channels have the following characteristics (Table 5.2): they are almost always phrased politely, follow a pre-defined schedule, are relatively wordy, tend to be one-to-many, and usually tell recipients nothing they do not already know. Formal channels, in fact, exist for the purpose of being an audit record; and while audit records are necessary and valuable, they are not often referred to while events are taking place.

In contrast, *informal* (or *implicit*) channels may be highly direct or infuriatingly indirect. They are profane, idiomatic, narrow, timely, and overwhelmingly tend to be one-to-one. Informal channels carry only important information. They are invariably honest. Rumour, belief, and folklore around Processes & Procedures are examples of informal channels.

Formal Channels	**Informal Channels**
are scheduled	as needed
are wordy	concise
mostly document what is already known	full of new information
almost always tactful	invariably honest
discoverable	deniable
politic	profane

Table 5.2: Distinguishing characteristics of formal and informal channels.

Informal channels carried the bulk of communication that contributed to progress in both the R' and R'' case studies. Management communicates

ostensibly through formal channels, but engineers behave differently; they communicate primarily through informal channels. Mills identified reasons for this nearly a century ago, when he studied the career path of scientists in industry; at first, he noticed that techies were different from managers, and not necessarily in an advantageous way:

> These men are driven by the instinct of workmanship and by an emulative desire for recognition by their peers. They are rarely aggressive and never acqusitive. They like to work on a basis of cooperative equality and are not of the type to become executives, even minor ones.
>
> With such characteristics they are particularly subject to exploitation [151, p. 92].

Mills observed that scientists—here considered as software engineers—exhibit a curious naïveté when it comes to their own needs:

> Labo[u]r, in its cap and overalls, bargains and fights; the upper levels of business twist and shift for profitable positions; but the scientific take what is handed to them, like wards of industry, instead of its chief benefactors [151, pp. 127–8].

The software engineers on R' and R'' exhibited these characteristics (as well as did the penetration testers, it should be noted); in short, they appear to have trusted everyone to be as honest as they were all the time, and while they write honestly in formal reports—because they write honestly in everything—the limitations of formal reports (especially timeliness) force engineers to communicate more and more through informal channels.

5.5.1 Characteristics of Formal and Informal Channels

Formal channels, by their nature, are highly resistant to disruption. The reason for this is because formal channels are transmitted through written reports that are scheduled in the project plan. When a new project manager

(PM) takes over, the first thing he or she does is to make sure that the next scheduled report is delivered on time. Reports are wide bandwidth but low entropy, as they mostly tell people things they already know. Formal channels in the guise of reports are extremely hard to disrupt, as oftentimes they are specifically listed in contracts, with due dates. Formal channels are re-established automatically when broken.

In contrast, informal channels are fragile. Re-establishing an informal channel requires negotiating trust, which is slow. Informal channels are narrow bandwidth and often high entropy (*i.e.*, they are encoded in a way that makes the recipient work hard to decode the meaning) and they carry with them lots of contextual information about the sender; decoding them depends on shared context; without it, oftentimes deliberately the message would be unintelligible to an outsider.

'Noise' in these channels corresponds to delayed reports,[24] incomplete or incorrect reports, delay of forerunner signals due to people being on travel, ill, or otherwise out of communication, and people not wanting to give bad news to their supervisors. 'Entropy' corresponds to the immediate usefulness of the information received. When placed against the inefficient but robust coding of formal channels, this justifies both the relative fragility and the practical necessity for implicit channels, thereby explaining their emergence in almost all cases.

Analogous to 'channel coding' in formal and implicit channels is the use by senders of phrasing, metaphor, or sarcasm. The latter, especially, requires additional work by the recipient to decode the meaning of the message, transmitting as it does a great deal of the sender's context along with the

[24]It is unclear whether latency, in this case, corresponds to a form of noise, or whether it is something else.

data.[25] The coding in formal channels is much more redundant and carries correspondingly less information.

Forerunner communication is the most immediately useful type of informal channel.[26] This was seen repeatedly in the R'' certification process; contrast this to R', in which almost no implicit channels were seen, although the formal channels were perfect. We conclude, therefore, that formal channels are necessary, but not sufficient.

There is essentially no noise observed in implicit channels. Forerunner communication—by definition, implicit—is the most valuable of information, but also by definition it is illicit (*i.e.*, out of compliance with process), and therefore sometimes must be encoded for security (of the transmitter) as well as for the available bandwidth, because implicit channels are narrow.

Secrecy is needed, sometimes, to protect the sender of the information, for when legal discovery is undesirable, in situations where some kinds of email or putting anything in writing is discouraged, the information exchanged through implicit channels[27] is much more important to a successful outcome than formal channels. From this we can conclude that the expected mode of communication of this type of information will be person-to-person; generalising, the amount of communication predicted is expected to be proportional to $\delta^{-\alpha}$ where the 'distance' δ between the sender and receiver, in this case, is the number of face-to-face hops, and α is in the range of 1 to 2; Odlyzko, in a review paper, calls this a *gravity model* [167]. Support for this argument—both efficiency and secrecy—can be found elsewhere in the

[25] There is a danger to this, however; formulated by Wiio in 1978 as his Law number 3: '*Jos sanoma voidaan tulkita eri tavoin, niin se tulkitaan tavalla, josta on eniten vahinkoa* ('if a message can be interpreted in several ways, it will be interpreted in a manner that maximizes the damage') [263], translated by Korpela in [134].

[26] Note that the absence of a signal *at the expected time* is a signal; it propagates at the same speed.

[27] *I.e.*, developer-to-developer, developer-to-tester, tester-to-developer, certifier-to-developer; and accreditor-to-accreditor, in the case of an SCI-like ST&E.

literature:

> '... such informal operation was typical of OSRD laboratories in World War II.[24] Rabinow, who has a gift for turning a phrase, over the years evolved a set of laws:
>
> > '[Law] #13 I think, says that everything you do illegally, you do efficiently. This, of course, is perfectly obvious. For one thing, you do not write at all because writing on an illegal project is suicide... [i]llegal projects are very, very efficient from many points of view. We were allowed to do much of this.[25]' [261, pp. 12–13] (notes in original).

Or, as Stephenson [219, Chapter 31] explained it carefully,

> '... two person conversations... are best'.

Spoken communications, for example the teleconferences in the R'' case study, lie between these extremes. On the one hand, they are usually scheduled, but—at least in the classified world—are never recorded and rarely minuted adequately. They are bracketed by implicit channels, however; the researcher observed numerous times participants in D arriving early for the R'' hotwash, informally conversing amongst themselves before the call, and usually discussing the outcome of the call after it ended. These conversations had the flavour of implicit channels from Table 5.2.

Reasons for the emergence of implicit channels in the CT&E activity now become clear: they include the inefficiency of required formal channels, the relative insufficiency of their capacity given typical schedules (which pressure directly leads to the emergence of implicit channels carrying forerunner communication in order to meet the schedule), and the emergence of channel coding for security of the sender of forerunner communication, the key for which is supplied by the difference between 'product' and 'process' knowledge.

Collins, *et al.* found implicit channels arising out of necessity when formal

channels failed to adequately transmit implicit knowledge:

> 'In this case, then, a search for inadequacies in the available 'algorithm' is not fruitful: information sources were untypically accessible, yet *deliberate* transfer of all the requisite knowledge was not managed' [54, p. 449] (emphasis in original).

Conway's observation, on the other hand, suggests that implicit channels may arise from the necessity for flexibility in engineering organisations tasked with designing arbitrarily complex new systems—something certainly reminiscent of the cross domain developer's problem—atop pre-existing formal lines of communication:

> ... organizations which design systems (in the broad sense used here) are constrained to produce designs which are copies of the communication structures of these organizations. We have seen that this fact has important implications for the management of system design. Primarily, we have found a criterion for the structuring of design organizations: a design effort should be organized according to the need for communication [63].

The fragility of implicit channels serves no direct purpose, but the fact of their existence in spite of the drawback of fragility argues for their necessity.

5.5.2 The Effect of Project Manager (PM) Changes on Implicit Channels

It is conjectured that discontinuities around changes in Project Manager (PM) tend to disrupt implicit communication channels more than formal ones.

The reason why changes in project manager disrupt formal communication channels less than informal channels are disrupted is because formal channels are operationalised in deliverables, which itinerant project managers—and their managers—track. Reports, considered as formal communication

channels between developers and project managers, project managers and programme managers, programme managers and government programme offices, programme offices and certification authorities, and—in the case of *ad hoc* reports—between developers and security testers, have the advantage of being written, visible, and referable, but they are not timely enough to facilitate optimal speed of reaction to events. Informal channels arise naturally whenever individuals are frustrated.

In retrospect, and further to the *post mortem* discussion at the end of Chapter 4, another reason we can now understand for the failure of the R' project was a distinct lack of implicit channels, caused in part by the fact that the project was so small—on D's side, the line organisation consisted of the project manager and J only: management and labour. Management and labour communicate through formal channels; the informal channels that J needed to be hooked up were to other software engineers, and as observed in R' (particularly with respect to the HLD and LLD), those channels were not open. Another useful destination for some informal channels from J to other engineers might have been to H or cert_23 at L, but in the eventuality, no such implicit channels ever emerged; in accordance with the contract between D and L, all communication happened through formal reports.

5.5.3 Fact and Belief: Reasons For Implicit Channels

> '... the misleading quality of some of the formal information available in 1972 can be seen in different laboratories' *beliefs and actions*...' [53, p. 58] (emphasis added).

This brings us to the difference between fact and belief, and the way to understand the emergence of implicit communication channels (Figure 5.11). The R' evaluation failed, among other reasons, for lack of adequate

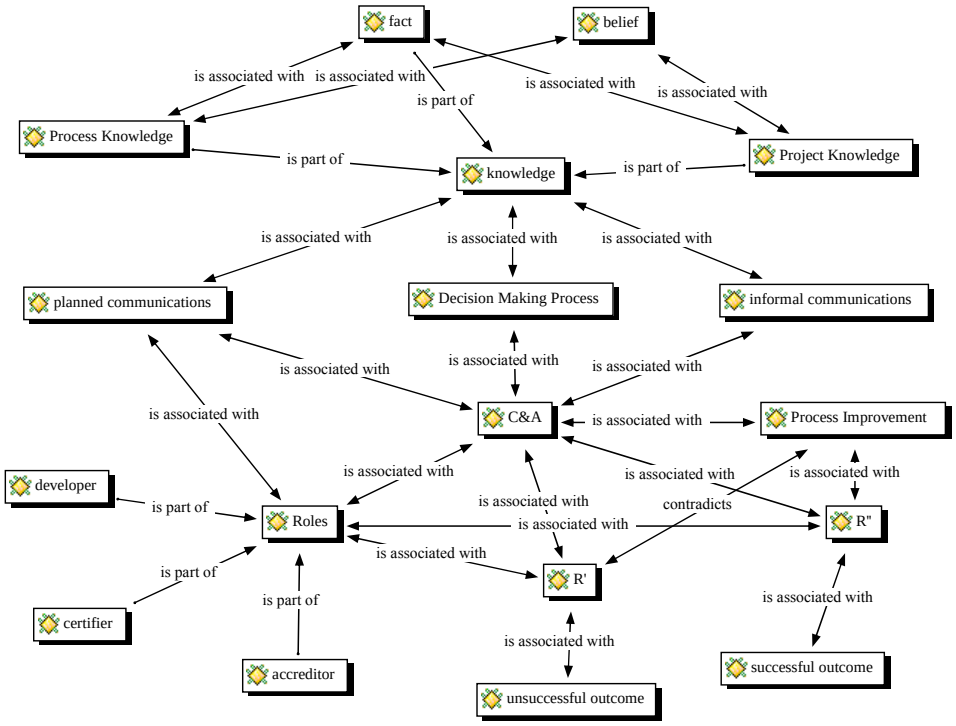


Figure 5.11: Grounded theory of formal and informal ('planned' and 'implicit') communication channels in the R' and R'' case studies.

training. As noted earlier, it is the difference between 'process' and 'product' knowledge that is the key to decoding implicit channels. Training provides the *process knowledge* needed to interpret facts; but belief and rumour are at the core of *product knowledge*. Training is one thing that was missing from the RTG 1.0 CC evaluation; the writers of the ST in Chapter 4 possessed instructions but no training:

> ... training gives us an understanding of our tasks and equips us to use our discretion, while instructions tell us precisely what we should and should not do; training equips us for knowledge-based behaviour while instructions equip us for rule-based behaviour. Which do we want?
>
> While admitting that most jobs today require some knowledge-

> based behaviour, many managers feel that in some situations, especially those involving safety, rule-based behaviour is essential. In practice, the rules never cover every situation and people have to use their discretion. They are then blamed either for not following the rules or for sticking rigidly to the rules when they were obviously inapplicable, another example of contradictory instructions... [t]o try to cover every situation the rules become more and more complex, until no one can understand them or find the time to read them... [i]t would be better to recognise that people have to be given some discretion, give them the necessary training, distinguish between the rules that should never (well, hardly ever) be broken and those that on occasions may be an accept that from time to time people will make the wrong decisions [129, pp. 52–53], referring to [237, pp. 120–128, 165, 175].

Process knowledge is needed for an individual to be able to use formal channels, but product knowledge is needed to interpret the implicit ones. Recall the developer's complaint about the 'arrogance' of N''s pentesters: is the criticism valid in light of what we know now about implicit channels? The [N'] employees' attitude of 'we are so much better than you' is clearly seen in an article from one of the agency's internal journals [186]. But the attitude perceived by D needs be re-assessed in light of recent disclosures; with regard to NSA, one could not realistically wish for a more technically competent—on the eavesdropping side, if not so well the protection-of-their-own-networks-from-privileged-insiders side—National Security Agency. The technical capabilities of N', objectively, surpass those of the developer. It provides a second example, and a verifiable one, of both the developer's limited view of the situation and of the asymmetry of knowledge inherent in CT&E. As explained by Starbuck, in the context of specialist law firms:

> Strategic-management theories say that expertise can sell for higher prices if it is valuable and esoteric—that is, generally unknown to the public and somewhat rare. However, when

> KIFs[28] deliver information and advice to clients, their expertise becomes explicit and partially exoteric, so KIFs are constantly at risk of destroying their primary assets [218, p. 1401].

In §6.3.1 we shall see an actual example of it, in the difference in operational philosophy between commercial and government penetration testers. The difference between fact and belief can even be seen exploited in the large: 'organisational façades' could be thought of as implicit communication channels on a grand scale:

> At Aerospace Corporation, an executive explained that his company had once had difficulty relating to its sole client, the US Air Force. Aerospace employed highly educated, highly paid engineers who had great autonomy in their jobs. Since these personnel had similar qualifications, Aerospace operated as a very flat hierarchy with only three levels. However, this flat hierarchy had caused problems for the Air Force officers who represented the company's client. Aerospace had been asking Captains, Majors, and Lieutenant Colonels all to interact with engineers at the bottom of the hierarchy, implying that Lieutenant Colonels were no different from Captains. Inconsistent role expectations caused confusion and tension during interactions between Aerospace personnel and their clients. After a time, Aerospace personnel realized that they should adapt to the expectations of their clients, and the company created a seven-level hierarchy for purposes of client relations. This hierarchy was actually a façade and Aerospace personnel ignored it during interactions within the company. Although Paul Nystrom and Starbuck (1984a) had written about organizational façades, this façade had a surprising degree of calculated superficiality [218, p. 1399].

Considered as an implicit channel, organisational façades exhibit many (though not all) of the characteristics in Table 5.2: (1) interpretation dependent upon shared context, (2) arising in the interest of efficient communication; and (3) the problem to be solved—an impedance mismatch, as it were.[29]

[28] KIF: Knowledge-Intensive Firm—see [217].

[29] The impedance mismatch between engineers and their managers was noted as early as 1951 by Moore and Levy [152].

5.5.4 Process Improvement

The R' security evaluation generated no process improvements and left D with nothing to show for the effort. In contrast the R'' certification's entire reason for being done (at the time and with the given participants) was to debug and validate the new process for CT&E of cross domain systems under ICD 503. Accordingly, a few links to the 'process improvement' code were seen (Figure 5.11), although more interesting than the number of links (for no explicit quotations relating to process improvement were seen; it was primarily an activity concentrated in the certifier, where it was out of sight) is the contradictory link to R'. Unlike in the second case study, visibility into process improvement effort was available in the earlier project, and the researcher can assert from first-hand knowledge that there was none captured at the time. This research, however, began as the attempt of a key participant in the R' failure to derive retrospectively the process improvement lessons available from R' before they are lost.

5.6 Discussion

As early as 1934, Popper wrote that scientists never prove their hypotheses, only disprove a null hypothesis; falsifiablility means that a hypothesis should make testable predictions. Economy facilitates this; the simplest hypothesis may—although not necessarily—be the quickest to prove false if it is incorrect [179, Chapter 7*n*]. We interpret this in the Occamian sense to mean that it should be the simplest theory that adequately explains the observations, or—using Box's terminology—has the *parsimonious* quality that 'residuals... are consistent with a white noise error' [35, p. 6*n*].

The model given here is phenomenological; it models only the behaviour

observed and does not attempt to explain *why* a participant acted in a particular manner in certain situations.

5.6.1 Limitations of the Model

The model described here has some obvious shortcomings. It is not quantitative; no attempt was made to bound the information transfer capacity of informal communication channels,[30] a longitudinal study ought to be done that would compare the relative incidence of formal and informal communication channels across similar and different kinds of projects, and measurement of activity—or even detection—of suspected informal communication channels when such may consist of vague impressions, body language, tone of voice, or intuition is problematic at best. As G.E.P. Box said of useful models,

> '... there is no need to ask the question "Is the model true?". If "truth" is to be the "whole truth" then the answer must be "No". The only question of interest is "Is the model illuminating and useful?" [35, p. 3].

Regarding detection and characterisation of informal communication channels, mention should be made of them in the context of data retention policies in engineering organisations causing non-compliance, another form of implicit channel. It was observed in the D organisation that wilful although clandestine non-compliance with the corporation's email retention policy was nearly universal, perhaps skewed slightly towards employees with longer tenure, although that was probably a result of experience combined with friendly under-the-table advice to new employees. What is certain is that

[30] Assigning an upper bound to the information transfer rate of *formal* communication channels is possible if (1) formal channels are limited to reports, (2) reports are limited to planned events associated with CLINs in the SOW, and (3) semantic content of reports is limited to text-only. Applying these constraints to the size and number of reports observed in the R'' certification phase only, the rate is slightly less than 0.8 bits/s.

savvy developers deliberately moved important emails from the mail server, individually or by automated rules, to private storage, often in obscure locations or with misleading file names. Developers did it to avoid *D*'s automatically enforced email retention policy. It was widely believed amongst developers that important information had been irretrievably lost on several occasions because of emails left on the mail server; many had experienced it personally. This led to the widespread perception that email server storage was unreliable.

Document retention policies exist as a kind of self-imposed, voluntary organisational amnesia. Reasons for them exist, of course, either explicitly because of regulatory compliance requirements or covertly because of fear of subpoena.

> 'This is why some investment banks save everything,' [as one commenter put it.] 'They create a rebuttable presumption that if they don't have it, it doesn't exist. (Often helpful when the other side alleges there is a "smoking gun")' [250].

In *D*'s annual corporate ethics training booklet, on page 11, the ostensible reason is implied to be '[to] maintain accurate business records' but it creates a dual situation of effective lacunar amnesia combined with near universal non-compliance with policy and procedures by people trying to do their jobs. Zawinski (1998) provides an alternative view; Netscape Communications had an email retention policy, it wasn't followed, and if it had been the outcome would have been better [271].

Most policies can probably be traced back to a person who got in trouble, or more precisely, caused trouble for the company. But email retention policies are different in their effect. Imposed from above, ostensibly for one of the preceding reasons, company officers evidently having decided that it is worthwhile in light of the obvious risks, perhaps count on the informal policies

of persons below them to mitigate the worst effects, knowing that sensible people will tend to keep copies of their important email correspondence. It establishes plausible deniability while retaining the option of sacrificing a scapegoat for violating policy should the need arise.

In an unconnected occurrence, D was once accused of mishandling Personally Identifiable Information (PII) in the guise of Passenger Name Record (PNR) data that had been provided to D by a customer. Either the customer had not properly maintained control of the PNR or the policy had been changed after the fact, but D was directed to purge the PNR data from its classified servers, at considerable cost to the developer, illustrating the fact that 'you don't always get to choose what PII you have on your network' [149]. Analogously, organisations do not choose the informal channels they get. Formal policies such as email 'retention' (meaning deletion) can damage—that is, after Shannon, to add noise to—formal communication channels (or even to the informal communication channels that seem to be the key to efficiency in the tasks that the organisation wants to accomplish) but they also cause compensating informal channels to come into existence. These channels, by nature, are undetectable and uncontrollable. As Backhouse and Dhillon (1996) said:

> Because security problems often arise where the communication necessary for co-ordinated action breaks down, our ability to model allows us to understand the communication aspects of security [12, p. 4].

In Chapter 2, we foreshadowed that penetration testers are stuck in Baskerville's (1992) first generation of using checklist methods exclusively; only the most advanced—generally, third-party independent security researchers and certain government departments'—penetration testers ever venture beyond use of prescribed tools and checklists for their methods [18]. There are

reasons for this, possibly persuasive ones, referencing a desire for consistency in security reviews and a certain minimum guarantee of test coverage (an attribute highly regarded by ISO 9000 proponents), cost, availability, and provisioning of trained personnel. But the bar set by most penetration testers, in the researcher's experience, is low. D's software developers are observed to be squarely in the second generation, with few forays into the third generation because of back-pressure from the government programme offices and certifiers, who as a rule distrust new methods—such as programming languages newer than ANSI C. The developer in these case studies tends not to employ checklists in the software development process, relying instead on experienced software engineers' judgement as to avoiding weak spots through thoughtful design—although this method incorporates the 'checklists' of the first generation implicitly, as embodying good well-known good design practices—which is, organically and without a specified process, the 'risk analysis' method of Backhouse and Dhillon's second generation. R'' was unique in the history of cross domain certification and accreditation because ICD 503 and DIACAP specifically 'use[d] risk analysis to identify the necessary [security] controls' [12, p. 2] and called for the beginnings of Baskerville's 'third generation' methods to be used in both software development and security testing.

5.7 Summary

The R'' case study led to the discovery of one grounded theory and the refinement of another. In the next chapter, the mechanism by which the developer of a cross domain system may for the first time exert a measure of control, however small, over the schedule—and hence the cost—of certification testing is described in detail. The theory of operation of that mechanism, grounded

in the data of meeting minutes of telephone conference calls and conversations with participants taken by an non-participant–observer with full access to the R programme and R'' source code, was discovered in the behaviour of D and N'. Another grounded theory, discovered years earlier from analysis of overlapping areas of responsibility in the R' project, was greatly refined and enhanced when it was realised during the R'' accreditation testing process that international accreditors are isomorphic to SCI accreditors who share no compartments.

Chapter 6

Interpretation

> *'Errors by managers are signposts pointing in the wrong directions'*
> [130, p. 207].

A useful model of accreditor behaviour was discovered in the grounded theory analysis of the R' evaluation and further refined by an insight from the R'' certification.[1] The accreditor model is interesting because it covers all of the possible cases of cross domain security accreditation, from simple two-sided collateral low-to-high accreditations to complex SCI and international accreditations where there exists no hierarchical relationship between two or more security compartments. It is useful to divide cross domain system accreditations into 'non–SCI-like' and 'SCI-like' categories. What is not at first apparent is that international accreditations, even if they are collateral, are SCI-like. A more powerful variation of the accreditor model is currently being developed that is believed to offer predictive capability to the developer, vendor, or installer of cross domain systems. Combined with the mechanism described in this chapter through which a developer can for the first time exert

[1] Portions of this chapter have been previously published in the proceedings of the *19th Computer and Electronics Security Applications Rendez-vous (C&ESAR)*, Rennes, France, 20–22nd November 2012, pp. 19–28 [143].

a measure of control, the accreditor behaviour model and the certifier control mechanism together offer the developer—at last—a means for forecasting and budgeting cross domain solution certifications and cross domain system accreditations accurately.

6.1 Reliable Signals

It can be shown that the cross domain system accreditor's problem is the same thing as the problem of market failure in the presence of asymmetric information familiar to economic theory. Market failures are thought to occur as a result of externalities associated with the goods or services being traded. In this case, the externality is that the total residual risk of the cross domain system acts like a 'public good' [28]. Furthermore the criteria for signalling established by Spence and Akerlof are met [1, 212]. This suggests a possible solution to the present high cost of certification and accreditation of cross domain systems that currently manifests in repeated testing and retesting of the same security criteria by accreditors at different security classifications.

6.1.1 Background

In the most general case, data owners nearly always mistrust one another, because the relationship between their security classifications may be non-hierarchical, or incommensurable, or simply equipotent, as happens in international installations [228]. Each data owner is represented by an accreditor or Designated Approving Authority (DAA) whose job it is to approve connection of the cross domain solution to a classified system or network and to permit operation for a specified period of time [161, 193, 244, 245]. DAAs work closely with the cross domain solution developer or installer and other DAAs to ensure adequate protection of the classified information in their

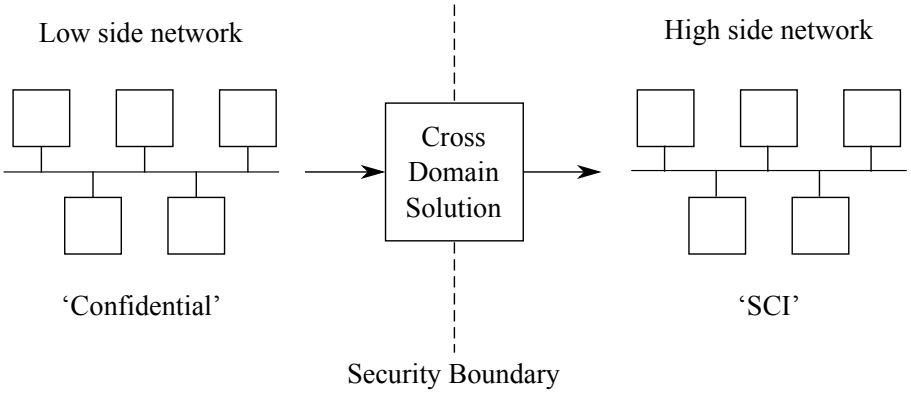


Figure 6.1: Simple cross domain system with asymmetric information

security domain. Data owners worry about two potential threats: accidental compromise of the confidentiality of classified information outside the security boundary (called a spill), and negative impacts to the integrity or availability of their information from the introduction of malicious code or denial-of-service attacks. DAAs, being people, in addition operate under the constraints of their government security clearance and security classification rules.

6.2 A Model of DAA Interactions Constrained by Different Security Clearances

Figure 6.1 illustrates a very simple example of an cross domain system that is nevertheless sufficient to elicit the problem.[2] There is generally no DAA for unclassified systems, so let us imagine that the low side is classified

[2]Other portions of this chapter have been previously published in the proceedings of the *13th IEEE International Conference on Technologies for Homeland Security (HST'13)*, Boston, Massachusetts, 12–14 November 2013, p. 103 [144].

Confidential and the high side contains Sensitive Compartmented Information (SCI). The low-side[3] DAA represents one of the military services because the information on the low side has a collateral classification, that is, it is classified but not protected by additional code words. But because the high side is SCI, which has a non-hierarchical relationship to collateral security classifications, the high-side DAA must represent one of the members of the intelligence community, for example the National Geospatial-Intelligence Agency (NGA). In reality, cross domain systems commonly are more complicated than this example, with multiple data flows in more than one direction, more than two endpoints, conditional information flows dependent on content, sanitisation, downgrading, and/or transliteration functions, and non-hierarchical security classification relationships. The model presented in this section, however, is sufficient to reason about cross domain systems in collateral, compartmented, and international installations.

In this model, DAAs having responsibility for information at different classification levels have security clearances and accesses that match their responsibilities. In the real world, that might not be true; all DAAs might be cleared for Top Secret/SCI. Our model presupposes the more limited case for two reasons: firstly, because it better reflects the intent of security policy irrespective of administrative convenience, and secondly because it allows us to analyse the important case of international CDS installations, where DAAs most definitely do not share mutual clearances.

Consider the following situation. DAA 1 holds a Confidential security clearance and has need-to-know, so is therefore privy to classified information about certain threats that are known to exist by the data owner of the low-

[3]By convention, cross domain solution developers habitually refer to data flows as being 'low to high' or 'high to low' despite the fact that the distinction may be a matter of opinion depending on the data owner's perspective.

side system. DAA 1 perceives a site-specific set of risks A_1 that affect the low side system, each risk computed from the *probability* of occurrence of an identified *threat* leading to an *impact* which is derived from the value of the asset [88].[4] Risks can be avoided, mitigated, transferred, or accepted [234]. DAA 1 assesses a set of risks based on the known threats at his or her clearance, the best available estimate of the probability of occurrence pT_i of each, and the value V of the information on the low side as perceived by the low side data owner; this set of risks that DAA 1 thinks it would be desirable to mitigate is:

$$A_1 = \bigcup_i [pT_i \times V] \tag{6.1}$$

where $0 \leq pT_i < 1$.

DAA 2 holds a Top Secret/ SCI security clearance with accesses similarly determined by DAA 2's need-to-know. It can be understood that DAA 2 knows about some highly classified threats that are not known to DAA 1. In practice, DAA 2 should be aware of all the threats that DAA 1 knows about, but this is not required by the model. DAA 2 perceives a site-specific set of risks A_2 affecting the high side based on probably a larger set of known threats, an estimate pT_j of their probability of occurrence, and the value V' of the information on the high side as perceived by the high side data owner. This is the set of risks that DAA 2 thinks would be desirable to mitigate:

$$A_2 = \bigcup_j [pT_j \times V'] \tag{6.2}$$

where $0 \leq pT_j < 1$.

A_2 is not necessarily a proper superset of A_1 or even needs be larger

[4] This is a very old idea; from 1662, 'Fear of harm ought to be proportional not merely to the gravity of the harm, but also to the probability of the event' is from [7], quoted in translation in [24, p. 71], citing [106].

than A_1. DAA 1 values low side information independently of DAA 2, and quite possibly assesses different probabilities for similar risks—although if they are seriously different, it might be better to treat them as distinct threats—simply because it is DAA 1's own asset on the line. Similarly, DAA 2 values high side information independently of DAA 1.

6.2.1 The Idea of Residual Risk

DAA 1 perceives a technology-dependent set of risks B_1 that it is possible to mitigate, and DAA 2 similarly perceives a set of risks B_2 that is is possible to mitigate. Because both sides are presumed to be aware of what is technically possible, it is likely that $B_1 = B_2$, although there is always the possibility that DAA 2 is aware of some highly classified risk mitigation for a threat that DAA 1 does not even know exists.

The job of a DAA is formally to accept responsibility on behalf of the Principal Accrediting Authority (PAA) for the *residual risk* of connecting a particular classified information system to the CDS. Residual risk for each DAA i is defined as the relative complement,

$$(A_i - B_i) \quad , \tag{6.3}$$

that being the set of risks which it is felt, by a particular DAA, to be acceptable not to mitigate. The goal of the DAA is always to minimise residual risk. This is achieved through a combination of choosing the right cross domain solution vendor and product based on Certification Test and Evaluation (CT&E) results, correctly configuring the cross domain solution according to its technical capabilities, and rigorous testing of the CDS before and after issuance of approval-to-connect to verify that the CDS adequately

protects the security domain of the data owner each DAA represents. In real installations, the PAA responsible for the highest-classification information in the system generally is responsible for choosing a cross domain solution vendor. The process of testing a cross domain solution *in situ* forming an XD is called Security Test and Evaluation (ST&E) and results, in the model, in the granting of an Approval to Operate (ATO) from each DAA. ATO lasts for a limited amount of time, usually three years, and is periodically reviewed.

6.2.2 Asymmetric Knowledge

To reiterate, by definition a CDS installation always spans at least two security domains controlled by different data owners. With multiple data owners come multiple DAAs. With each DAA, under present rules, comes another round of ST&E, oftentimes performed by the same team of Independent Verification and Validation (IV&V) contractors for reasons described in [142].

It is from the asymmetry of knowledge just described that the well-known time and cost inefficiency of the CDS accreditation process arises. If the true level of residual risk could be agreed upon by all DAAs and validated by a single round of ST&E to the satisfaction of all parties, then the cost of CDS accreditation would be greatly reduced. Towards that goal, we now show that the problem is isomorphic to a well-known result from economic theory.

Problems that can be caused by asymmetric information are well understood. In markets characterised by a lack of knowledge on the part of buyers, rational behaviour by all participants can lead to a collapse of the market to the point where no seller will offer a product for sale [1]. Conversely, in markets characterised by a lack of knowledge on the part of sellers, *adverse selection* results in a lopsided distribution of risk, which can lead to a situ-

ation called *moral hazard* in which participants who know they are insulated from the consequences of a risk behave differently than if they were fully exposed to it [68]. Game theory offers a handful of compensating strategies for asymmetric knowledge, among them the concept of *signalling* [212, 222]. In signalling, sellers in a market under conditions of asymmetric information can resolve the asymmetry by communicating information to buyers in a convincing way, but in order for the buyer to believe the signal, the cost of asserting the signal must be high [212].

Can we apply these ideas to the problem of improving the situation of a temporary non-optimal equilibrium amongst the individual assessments of n different DAAs about the total residual risk resulting from the installation of a complex CDS? In one sense, the problem is that non-communicating DAAs can end up stuck in isolated local minima because they lack an important piece of information about a risk mitigation already implemented by another DAA in response to a threat the existence of which is above the first DAA's clearance level.[5] In another sense, the problem is analogous to a covert channel, through which we wish to communicate some information in violation of the system security policy [157]. In that case, the security policy we need to violate is not that of the CDS, but of the security clearances of at least some of the DAAs. The requirement is not entirely dissimilar to the establishment of a *subliminal channel* [210, p. 459] except that there is no encryption involved; therefore, we deduce the existence and some of the characteristics of a missing component that would enable solution of the problem: there would have to be at least one randomly chosen value exchanged in the protocol [211]. (See §6.3.2.)

[5]The related problem of highly classified threats for which there is no known risk mitigation is a very real one, but in the absence of a fix, from the perspective of the higher-cleared DAA it is a worry he or she cannot talk about, and from the perspective of the lower-cleared DAA, ignorance is bliss.

6.2.3 Justification for the Accreditor Model

Is it even meaningful to talk about a single value for the residual risk of a complex CDS interconnecting many different security enclaves, thereby exposing data of widely differing perceived—and maybe even objectively intrinsic—values to the risk of damage, disclosure, or loss? It is attractive to call the overall residual risk

$$R = \bigcup_{i}^{n} (A_i - B_i) \tag{6.4}$$

from the residual risks in (6.3) assessed by each individual DAA—who is, after all, responsible for the safety of data in his or her security enclave—because this metric behaves the right way in the intuitive sense that if one DAA feels that the residual risk to one enclave is unusually high, it properly increases the overall level of risk of the CDS.

It is claimed that this model is sufficiently powerful to address every situation encountered in the field. To show this, first consider a collateral CDS where each accreditor has a security clearance that is one of confidential, secret, or top secret.[6] The highest security clearance of any accreditor in the system, and consequently the security classification of the CDS, is called 'system-high'. The lowest security clearance of any accreditor in the CDS, in the model, determines the classification of 'system-low'. In a collateral CDS, each accreditor's security clearance is hierarchically related to all of the others such that when any accreditor cleared at system-high is satisfied that the residual risk is acceptable, all the accreditors immediately agree because they know that the system-high accreditor already knows everything

[6]The presence of uncleared accreditors, who can be considered to have a clearance of 'unclassified', such as might be represented by private organisations with information security responsibility such as health care providers, does not invalidate the relation.

they know about the threats and vulnerabilities of the CDS.

Now consider the case of a CDS containing SCI. Here there is no strictly hierarchical relationship between the security clearances of accreditors, in practice some of whom might have collateral clearances. System-high floats to SCI (which dominates all collateral classifications) with the union of all applicable compartments; the definition of system-low remains as before. Now if there are any accreditors who are not cleared for SCI, or there are at least two SCI-cleared accreditors who do not share at least one compartment, we are at an impasse—at least one accreditor may know of a threat or risk mitigation that affects the residual risk of the CDS but is prohibited from communicating that information to at least one other accreditor. In this asymmetric information situation, only a limited amount of information can be legally communicated without violating clearance or classification rules: the fact that a particular accreditor believes the residual risk of the CDS to be too high.

6.2.4 Information Leakage

Interestingly, this leaks information; recall the example given earlier of an accreditor who is cleared to know about a highly classified threat or risk mitigation. There are only three possibilities: firstly, if the accreditor says only that the residual risk is unacceptably high, that statement reveals two facts: the existence of a classified threat and the fact that no one knows how to mitigate it. The second possibility is if the accreditor says only that the residual risk is acceptable; this leaks the fact of the existence of a classified risk mitigation, although not necessarily the fact of a higher classified threat, as the classified risk mitigation may be for an already known threat with no known risk mitigation. It is an unavoidable leak, however, as the only other

Type	Security Clearances	Special Characteristics
1	collateral	simple hierarchical lattice
2	SCI	non-hierarchical
3	international	non-hierarchical and non-comparable

Table 6.1: Types of accreditations

alternative would be for the accreditor to remain silent, thereby accepting personal responsibility for a risk that is intolerably large. The accreditor must say something.

The conventional wisdom holds that security rules ought to be strictly enforced. If the Bell and LaPadula security policy is followed exactly, as in the assumption that accreditors are cleared only to the level they need to be, then the security policy must be violated under some conditions. If, however, the security policy is relaxed slightly—for efficiency—to allow having a pool of accreditors all cleared to the highest level, and any accreditor can work on any accreditation, then the problem goes away. To improve security and close a covert channel, relax the security rules—this is counter-intuitive.

No solution is known for this problem. It logically derives from the model. The scenario, however, is real and drawn from personal experience of the researcher with cross domain system installations in the field.

6.2.5 International Accreditations

Finally, consider the problem of international accreditations (Table 6.1). Without loss of generality, international accreditors can be treated as SCI-cleared accreditors who have access to only one compartment, that compartment being the name of their country. (The uncleared accreditors mentioned previously could equivalently be modelled as foreign country accreditors.)

This is consistent with the extension of collateral classifications with handling caveats such as NOFORN ('not releasable to foreign nationals') or EYES ONLY. The model is therefore complete.

6.2.6 Part 1: Predicting the Behaviour of Accreditors

The traditional view of signals holds that for a signal to be convincing, it must have a high cost to preclude dishonest use of signals to gain unfair advantage [212]. It is believed that Spence's cost constraint is satisfied in this adaptation of the model because there is negative incentive for cheating when the result of dishonesty—that is, to communicate a false reading of the residual risk as perceived by DAA k—would either raise the value of R in equation (6.4), thereby increasing the amount of risk that DAA k must accept formal responsibility for, possibly even to a level exceeding DAA k's authority; or conversely, to artificially depress the apparent level of risk below what DAA k knows the true value to be, again raising the level of personal risk to DAA k's own self when he or she signs on the dotted line.

The required high cost of signals in this market is made manifest by the very real risk that a cleared DAA takes in choosing to communicate information about the true level of residual risk in violation of his or her oath to protect classified information. It works both ways, as even DAAs with low security clearance understand the need-to-know rule and would hesitate to casually provide classified information to another absent a clearly communicated need-to-know decision from their authorised security officer. The necessary and sufficient criteria for signalling, therefore, are met [1, 212, pp. 499–500 and 367, respectively].

6.2.7 Part 2: Controlling the Schedule of Certification

The developer cannot accurately forecast the schedule of certification testing during CT&E because the duration of the penetration testing phase is unknown. Very much like covert channel analysis in high-assurance Common Criteria evaluations, penetration testing tends to be unbounded in the possible effort that could be expended and always is effectively terminated either by funding or schedule, not by the completeness of testing. Good pen testers—and the N' ones in Chapter 5 were good—never stop. Pen testing and CCA are rabbit holes of indefinite depth[7] and invariably are stopped when a certain amount of money and/or time have been spent, not because testers ran out of ideas. Given unlimited funding, they would never stop, the schedule would fail to progress past Beta I, and the system would never be installed. Like the 'experimenters' regress' described in Collins [53, Chapter Four], penetration testers have no way of knowing when they are finished:

> '. . . [w]hat the correct outcome is depends upon whether there are gravity waves hitting the Earth in detectable fluxes. To find this out we must build a good gravity wave detector and have a look. But we won't know if we have built a good detector until we have tried it and obtained the correct outcome! But we don't know what the correct outcome is until. . . and so on *ad infinitum*.[8]

It was accidentally found during the R'' certification that the cross domain solution developer can indirectly control at least the duration of penetration testing. A causal correlation was observed between certifier findings reports and the form of the developer's responses (see Figure 5.8 on page 124). When

[7] This was proved in 2010 by Böhme and Félegyházi under the iterated weakest link model [30, 31].

[8] Consider the parallel between pen testing—or CCA—and Underwriters Laboratories (UL) ratings of safes and vaults for burglary resistance; safes are rated up to Class 3, which promises only 120 minutes before a competent safe-cracker, who is assumed to have the proper tools to hand (but only hand tools—no explosives or thermal lance), will breach the vault [241].

the developer responded by disagreeing with the findings of the certifier's penetration testers, this invariably prompted a follow-on report containing more findings. When the developer concurred with the findings, no further reports of findings appeared. It is put forth that this mechanism may be usable by the developer to bound the schedule of certification.[9]

It might be argued that the certifier control mechanism is simply human nature. That it is, but it appears to be a reliable control nevertheless. In the absence of any other available control mechanism from the developer to the certifier, and especially to the penetration testers employed by N', it is valuable.

In general, accreditors might exchange information or not. If they exchange information, nothing more can be said. If they do not, it could be because they do not have any information to exchange. Or it could be that they have information, and would like to exchange it but are prohibited by security clearance or security classification rules. Or they might not want to exchange information.

6.3 Further Thoughts

The most visible consequence coming out of analysis of the case studies is that apparently intractable cost and time overruns in certification and accreditation of cross domain systems arise from repeated re-testing of the same or similar test procedures,[10] by the same people,[11] demanded

[9] It should be noted that it is possible that the funding source of the certifier's penetration testing effort was looking for a chance to gracefully bring penetration testing to a close without seeming to have yielded to the wishes of the developer.

[10] The test procedures are usually written by the developer, being the one best qualified to write test procedures. Since the developer must have already written test procedures anyway as part of the software development life cycle, re-use of the software developer's test procedures ensures an irresistible cost savings to the government programme office.

[11] It is most common for the test procedures to be run by the same team of IV&V contractors each time, again looking like a cost savings to the data owner because they are

by mutually distrustful data owners at different and often non-comparable security levels that are inherent to any cross domain system installation.

Furthermore, the tenure of project managers, empirically, seems to be key. It was an identified factor in the R' case study and was specifically called out in the recommendations of the FiReControl auditor's report amongst other places [155, 90, 258]. Observationally it became clear—and this is validated implicitly by the R' case study as well as explicitly from other sources subsequently—that it was project manager tenure, surprisingly, that seemed to be most strongly correlated with certification success. This was a complete surprise that came out of the grounded theory analysis. It is thought unlikely to have been noticed any other way.

Consider what went wrong in the R' Common Criteria evaluation from the perspective of Conway's thesis: 'organizations which design systems are constrained to produce systems which are copies of the communication structures of these organizations' [39, 63]. Often-times oversimplified when separated from the original paper to 'if you have four groups working on a compiler, you'll get a 4-pass compiler' [185], it is no *bon mot*; Conway demonstrated convincingly that the homomorphism between system designs and organisational structures arises necessarily from communication channels in every organisation that follow administrative reporting relationships, themselves set as a result of empirical experience [64, 262].

The organisational structure at the time of the R' validation went from the Programme Manager to the Project Manager to the author of the ST, with an intermittent cross connection from the ST author to L via the testing laboratory's technical representative H. What was intended to be produced was a complete set of work packages including an ST, HLD, and LLD without

already trained on and familiar with the system.

benefit of a PP. What was actually produced was an overly large ST and an interesting design extraction tool that produced part of an LLD and part of an HLD. Almost everything else that was delivered to the validator was pre-existing documentation. None of the many project managers on R', on either the D or L side, detected the problem, and it was never corrected.

In contrast, during the R'' DIACAP certification there was almost no turnover of project managers (or software engineers), although there was a large and ever-shifting population of IV&V contractors, regression testers, pentesters, certifiers, and funding agencies. DAAs, it has been observed over nearly twenty years on the R programme, do not change often.

R'' worked because the project was continuously monitored and controls adjusted frequently. But there is another parallel that can be argued, tying the families observed in groupings of codes in the ATLAS.ti diagrams in Chapters 4 *and* 5 to the grounded theory of why R' failed but R'' succeeded: the effect of incompatible organisational maturity levels at the interface between developer and certifier.

The R' developer was 'CMMI Level 1' with respect to the Common Criteria; they didn't know they needed a process specifically for the purpose. L may or may not have had one; regardless, D was unable to use it. But there is another axis, namely the dependency or familiarity of a developer with its policy–process–procedure-oriented environment (Figure 6.2). As we saw in Chapter 5, mature organisations develop policies, processes, and procedures in response to errors made in the past, or lessons learnt from watching the fate of others. Put on a quadrant chart, we have R'', the successful one, in Quadrant I and R', which failed, in Quadrant II. R' failed because L did not present a process that was compatible with D's policies and procedures. The normal situation, that CCTLs are used to dealing

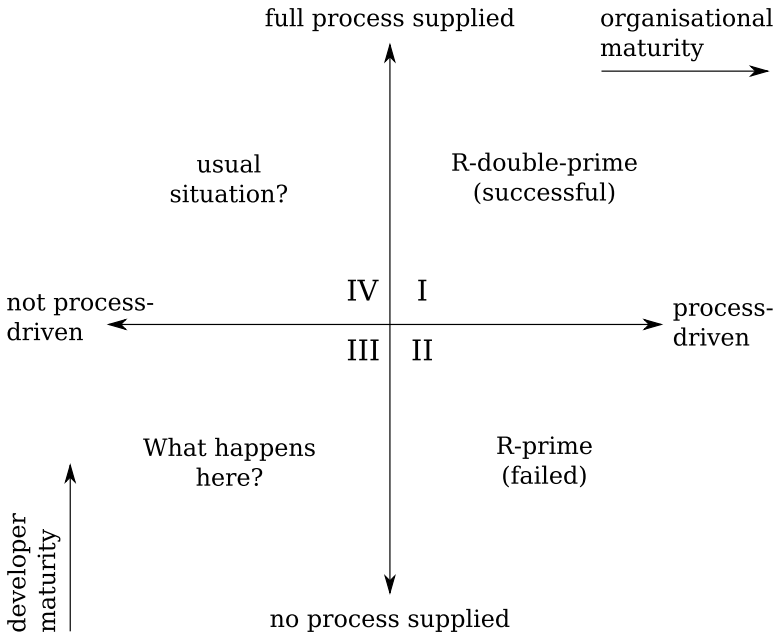


Figure 6.2: Quadrant I is the situation R'' found itself in: a process-driven organisation encountered a well-defined process and this led to success. The failure of R', located in Quadrant II, was due to a process-driven organisation encountering a situation where there was no good process defined (L, in this case, did not supply a well-defined process). It is thought that the more common occurrence happens in Quadrant IV, where a well-defined process exists to guide *any* organisation, even one that fails to recognise it needs a process. What happens in Quadrant III remains a mystery, because like Abraham Wald's allied bombers over Germany in WWII, some projects never made it back to base [147, 254] (see §6.3.2).

with, is in Quadrant IV, where the developer comes to the CCTL with a product but no preconceived notions—the developer has no process for security certification, they have never done Common Criteria before, and they only know they need a certificate of evaluation to be eligible to bid on contracts. In Quadrant IV, the CCTL brings a well-defined process, and the developer follows it; an example would be the Common Criteria certification of the OWL data diode [95, 96, 115]. What happens in Quadrant III is less clear. Without a process supplied, we can speculate that a non-policies-and-procedures-oriented developer might logically attempt to mimic the *visible* portions of evidence of successful certifications, that is, evaluator shopping, EAL and TOE gaming, and published certification reports. With the number of in-evaluation products no longer published as of October 2014 [Welke, *op. cit.*], we are left in a situation of trying to draw conclusions from no data (see §6.3.2).

6.3.1 Misunderstandings Between Developer and Certifier

One persistent area of misunderstanding in the R'' certification was the professional security testers' implacable insistence on the existence of a particular (classified) theoretical attack against R. In D's view, it was an unimportant issue because D mistakenly at the time believed the vulnerability to be un-exploitable in practice. The developer was puzzled by the government security testers' repeated finding of the identical issue in consecutive certifications. 'In what possible way', wondered the developer, 'could this capability ever be useful to an attacker?' The reason for the disagreement between D and the penetration testers is understandable in hindsight—the classified vulnerability made sense only if thought of as a covert channel. And R had

never had a Covert Channel Analysis (CCA) done.[12] The developer never fully appreciated the finding, even though it was repeatedly put forth by different security testing groups in N' and N, because the threat model was never stated, only implied. The developer lacked a person with the correct *einstellung* to interpret the implication and the government never pursued the question of why their repeated urgings were never addressed properly other than by vague responses from D of 'that doesn't make sense', instead of 'show us your threat model'. It is conceivable that, if asked, the government would have refused to divulge the threat model on the basis of security—a further example (see Chapter 5) of professional security testers wanting to maintain their stock of knowledge against leakage, necessitating replenishment requiring original research—but the government people were never asked because the developer failed to think about the problem in the right way.

Every time pen testers write-up a finding and give it to the developer, the action unavoidably leaks information about the pen testers' capabilities. This creates a tension in the pen testers, who want to find vulnerabilities but don't want to give away their secrets—because the pen testers have to continually replenish their stock through either purchase or original research.[13]

[12] The reason why R had never had a CCA is because when R was designed and approved for use under DCID 6/3, it was approved for use at Protection Level (PL) 4. PL-5 was never seriously considered in part because PL-5 necessitates a CCA.

[13] The contrast between the *modus operandi* of government and commercial penetration testers is striking but not widely noticed (because of the small overlap between the respective populations of government and commercial penetration testers on the one hand, and of classified and unclassified software developers on the other). Commercial pen testers—and civilian contracting firms working for the government—for the most part, use tools like Nessus and `nmap` from a standpoint of unprivileged nodes on the network, to exploit known or suspected vulnerable services and ports. Government penetration testers characteristically tend to use static code analysis, debuggers, and disassemblers in a much more sophisticated and low-level attack mode on executables and on processes in memory, to find previously unknown vulnerabilities that the commercial tools have not been extended to know about yet, in essence, employing bespoke 0-days in a perfect example of the husbanding, destruction, and replenishment of esoteric knowledge described by Starbuck in §5.5.3 and [217].

6.3.2 Shortcomings of This Research

The accreditor model, like most theories in microeconomics, depends on a few unrealistic restrictions on human behaviour; it may be noted that real world behaviour is actually closer to the 'efficient market hypothesis' than is common with such theories—a reversal of the usual situation!

For reasons of cost already belaboured, it would have been desirable but was impossible to examine more than two case studies before drawing conclusions; validation was primarily anecdotal, for the same reason. Are the effects discovered real? The question cannot be answered without measurement of the false discovery rate, which is meaningless in the absence of information about the specificity and sensitivity of the tests, incidence of project manager turnover in the population—which could be discovered—and the incidence of failure in projects of comparable type [55]. That, in turn, in the context of software engineering research, depends on the conduct of more case studies; and those, as noted, are rare occurrences and expensive to access.

There is a danger in drawing conclusions only from successful projects; *survivorship bias* and the problem of non-ignorable non-response can lead to missing important lessons from 'the ones who didn't come back' [147]. Wald set the standard for operations research with his battle damage study, making specific—and often counter-intuitive—suggestions for improvement from data induced to be missing:

> . . . [t]he greatest probability of being destroyed is .534, and occurs when a plane is hit by a 20-mm cannon shell on the engine area. The next most vulnerable event is a hit by a 7.9-mm machine gun bullet on the cockpit. These, and other conclusions. . . can be used as guides for locating protective armo[u]r and can be used to make a prediction of the estimated loss of a future mission [254, pp. 88-9].

Pfeffer, in 1981, said:

> 'The population ecology perspective of Hannan and Freeman (1977) takes organization theory to task for studying only surviving organizations. They argue that selection on the dependent variable, in this case, survival, diminishes what we can learn about organizations. Rather what is needed are studies that incorporate both organizations that survive as well as those that fail, to understand what determines survival or failure [177, p. 410].'

The grounded theory approach, as previously argued, is nearly ideal for the purpose of software engineering research, as it effectively allows one to run experiments retrospectively on existing sets of data. The problem with existing data, though, is that sometimes it is less than perfectly collected. The R' data was poor, compared to the R'' data, and the R'' study would have benefited from a larger number of formal interviews with participants—especially with accreditors—that alone might have yielded more reliable quotations. But, as was found, certifiers and accreditors are chary of talking to researchers, and only one (ivv_2) agreed to do it. The repeated re-transcription of notes from meetings—from classified to unclassified to export-controlled to anonymised—probably contributed to some loss of information.

Connection Between Theory and Analysis

> All models are wrong, but some are useful.
>
> —George E. P. Box [35, pp. 2–3], also in [33] and in [34, p. 74].

Stipulating the poor-to-variable quality of the data, we have already claimed that the theory is useful. The execution of the methodology by which it was arrived at, however, was not as good. The researcher was unskilled in grounded theory methodology, and initially resistant to using it. The form of the data in R'' was not ideally suited to the ATLAS.ti tool in two respects: firstly, that tool is oriented toward interviews with subjects,

transcripts, and verbatim quotations—none of which could be obtained from a classified project, cleared individuals, and meetings that took place in a SCIF. Secondly, a much smaller number of records survived from the older project, R', which (because it was international) was even more stringently classified. Originally, it was intended to use R_0 as a third case study, but almost no records survived from that project, and only one of the participants (see footnote on page 79 in Chapter 4).

The grounded theory methodology was not fully carried through. There is a gap between coding and analytical memoranda that shows up in the absence of *categories*. The researcher had a difficult time with this step, and it delayed the production of Chapter 5 for a long time. Finally it was realised that the missing introspection—all of the analytical memoranda that should have been in ATLAS.ti—already existed, but in another place: the researcher's regular weekly activity reports. For years these reports, some of them thousands of words long, had been discussing the observations, interpretations, hypotheses, predictions, and theory with the researcher's supervisors. Once these were found and anointed as analytical memoranda, and after a couple of meetings with an experienced grounded theory methodology user (Dr Fléchais) who drew the researcher out and showed that a theory existed in these interpretations, the thesis rapidly proceeded to completion.

Compromised Results

How then, specifically, is the theory connected to the data, if not (completely) through ATLAS.ti? In other words, can we rely on it? The failure of R' in Chapter 4 was blamed on two things: short tenure of project managers—implicated by several other authors in the literature—as well as overlapping areas of responsibility, and a processes-and-procedures–oriented developer

operating without a procedure. Only the last cause was discovered using network diagrams in ATLAS.ti; the other conclusions, and the quadrant chart in Figure 6.2, preceded its use.

The diagrams in Chapter 5 were crucial to discovering the existence of planned ('formal') and unplanned ('informal') communication channels between the developer and certifier in R''. The clue to the existence of these channels was the obvious partitioning of groups of participants in the ATLAS.ti diagrams by their interactions; the researcher began to wonder what information crossed the interfaces. Coding of participants' roles made it obvious, in light of earlier conclusions, that the tenure of project managers was entirely different in the project that succeeded. The behaviour of developers and pen testers in Figure 5.8 was not discovered using ATLAS.ti; it was found by looking at the sequence of reports and responses in those subgroups of participants. The accreditor model was found independently of the data collected from R' and R'' because the researcher started drawing Venn diagrams of all the historical installations of R when thinking about the problems faced by cross domain system installers in Chapter 1. Those diagrams do not appear here, but are fully explored in reference [144].

Methodological Implications for Future Researchers

Three lessons were learnt: firstly, to store everything in ATLAS.ti (or another equivalent grounded theory tool). If the weekly activity reports had been tagged as analytical memoranda in ATLAS.ti, then their contents would have been indexed by the auto-coding utility and linked to, highlighting their importance earlier. This would have saved a year and a half.

Secondly, interviews are extremely important. Despite grand plans, only four interviews were successfully conducted with participants (K_1, K_3, T,

and ivv_2; although serious attempts were made to interview cert_20 and cert_6, they refused). This is a risk of studying participants who don't want to lose their security clearances.

Thirdly, the problem was too large. A viable thesis could have been written on the accreditor model alone, or the failure of R', or the care and feeding of informal channels. These were all so interesting, and so clearly important to tell the world about, that they were explored and included at the expense of timely finishing. These were not even blind alleys; there were a few of those explored fully too, but that was expected.

Validation is almost completely lacking in this thesis. It is claimed that the accreditor model points the way to establishing a controlled communication channel between accreditors. Such a channel, it is said, would allow them to agree on the true level of residual risk without violating security policy, and without over-spending on unnecessary repeated retests. In truth, how to do it is not yet known. In the discussion of subliminal channels, it was predicted that the notional channel protocol must include a random value, but we have no idea what that random value is represented by.

Without validation, the results here comprise some interesting logic puzzles and a few suggested guidelines for project managers. Certification events for cross domain solutions are rare and expensive; that makes developers and certifiers highly value new techniques that might reduce cost (or improve security), but opportunities for validation are scarce. The risk and cost associated with trying unproved techniques may make uptake slow.

Without replication, these results are not science. 'Scientists must write', said Barass [17], and the necessary condition for replicability is satisfied with the publication of Appendix B. But the sufficient condition depends on more case studies, more failures, and more successes. However, the research

was done with what was available, because it was important that somebody should do it.

The reason is because residual risk is of *critical* importance to DAAs—and because of the inevitable knock-on effects of DAA nervousness—to CDS developer fortunes as well. Sometimes it simply is not possible to know what the actual level of residual risk was, even in life-critical applications:

> 'In addition to their degraded actuators, the orbiters have aging plumbing lines leading into the main engines that could prove troublesome. Years ago, routine inspections revealed cracks in the flow liners of the feed lines, which carry liquid hydrogen from the external tank to the shuttle main engines. "If they had broken off, they could have gone into the engine and caused a catastrophic failure," says Seriale-Grush.
>
> 'NASA replaced the flow liners, but had no way of knowing if there were cracks higher up. While the orbiters were still flying, removing the 12-inch diameter feed lines would have cost millions of dollars, an unjustifiable expense considering there was no evidence of damage. It would be like removing a person's colon to see if there was cancer, just because the patient was getting old.
>
> 'Borrowing a page from modern medicine, NASA decided to go another way. Technicians threaded borescopes into the shuttles' plumbing to look for cracks. Technicians performed these fuel-line colonoscopies regularly, searching for the tiniest signs of damage. The examinations kept the shuttles flying, held costs down, and minimized risk. *Studying the feed lines now will show just how much risk remained*' [91, pp. 28–29 (emphasis added)].

Space Shuttles were launched more than a hundred times before the actual level of residual risk in that system was known.

Chapter 7

Summary and Conclusion

> *'Never remove equipment before you know why it was installed. Never abandon a procedure before you know why it was adopted'* [128, p. 22].

As David Bell put it in his re-visit of the MULTICS paper, 'In our real-world environment made up of multiple single-level networks—that is, relatively isolated networks each of which comprises a security enclave or domain at a particular security classification—connected to the network cloud, it is often necessary to move information across security boundaries, and by the Intermediate Value Theorem for Computer Security (CS-IVT), at least one multi-level component must exist in the cloud' [20, §6.2]. Cross domain systems are not only commonplace in classified networks, they pop up in hidden form almost everywhere.

To be able to predict accreditor behaviour is very important. Accreditor actions, and to a lesser extent those of certifiers (because accreditations happen more often than certifications) are the primary limiting schedule factor in the overall cost of cross domain solution installations.

Certification costs are highly variable because certifications are relatively

rare events and depend strongly on functionality amount and functionality changes, but generally exceed two or three times the cost of a typical installation and accreditation, more in time than money. The cost of an accreditation today approaches that of the hardware and software combined.

With this thesis, the beginnings of a useful model for explaining accreditor behaviour are in hand. A more general model of accreditor behaviour based on gradients and expected to improve predictive power is in development, but not well understood yet. A weak method for control of certifier behaviour by the developer during cross domain solution certification testing is also in hand. Both the accreditor behaviour model and the developer's control mechanism for certifier behaviour are hoped to assist in doing both certification and accreditation better next time.

It is widely acknowledged that information security Certification Test and Evaluation (CT&E) is expensive in terms of time, number of people required, and duplicated effort. That high cost makes CT&E a special event that is not done often, either because fewer systems requiring CT&E are made, or because not all of them are tested, or because the ones that are made are tested less often.

Interpretationally, from an insight that certifiers and accreditors tend to conflate the principle of defence in depth with the practice of independent verification and validation, a connection was found to economic theory and a minor result was proved showing that Spence's criteria for signalling are met [222, 1, 212, 223]. In future work, a prescription for minimising the repeated testing and re-testing of the same or similar test procedures by mutually distrustful data owners is believed to be solvable by means of controlled communication channels[1] without violating the global security policy.

[1]That is, communication channels having some of the characteristics deduced in §6.2.2.

7.1 Contributions

This thesis makes the following contributions:

7.1.1 Environment of Untrust

It is an unappreciated fact that the difficulty of cross domain solution certification and cross domain system accreditation lies inherent in the fact that—by definition—they always span at least one boundary between security domains controlled by mutually distrustful data owners.

Understanding this fact, and the parallel truth that cross domain system developers, vendors, and installers are continually faced with a certification and accreditation environment in which the rules are forever changing, is key to solving the problem of the high cost of CT&E and ST&E.

7.1.2 Conflation of Principle with Practice

People involved in the certification and especially the accreditation process conflate the duplication of effort presently happening everywhere in the practice of cross domain solution and system independent verification and validation with the security principle of defence in depth, which it is not. Yet the way certification and accreditation programmes are structured presents the appearance of irresistible cost savings to managers. The resulting overexpediture of time and effort yields no attendant increase in security assurance.

7.1.3 Accreditor Model

Even more duplication of effort arises from the structurally limited view of the cross domain system that is to be had by accreditors with different

security clearances and responsibility for information at different classifications because the accreditors are constrained from communicating freely with other accreditors in order to agree upon the true level of residual risk.

The key to solving this problem is to understand the explicit and implicit communication channels that exist amongst accreditors at different security levels, and between developer and certifier, through which people arrive at a shared comprehension of the level of residual risk in a cross domain solution certification or a cross domain system accreditation without violating the global security policy.

Towards this end we present a model of inter-accreditor communication that is sufficiently general to reason about cross domain systems interconnecting security classification levels that are both hierarchically and non-hierarchically related, *i.e.*, powerful enough to handle international accreditations. Relatedly, we derive a model of certifier behaviour.

7.1.4 Validation

The symmetry of having an accreditor model that predicts behaviour in the shorter but more often performed ST&E phase of cross domain system accreditation and a certifier model that admits a measure of control over the schedule in the much longer but seldom done CT&E phase of cross domain solution certification is pleasing but serendipitous. The implicit communication channels predicted by the accreditor model are proved to exist in the real world, and they satisfy the criteria for reliable signals of Akerlof, Spence, and Stiglitz.

The accreditor model shows up a flaw in current official government security policy; it is conventional wisdom that, to avoid as far as possible the inevitable 'ratcheting up' of security classifications in the Bell–LaPadula

model, all personnel should be cleared to the lowest security level consistent with accomplishing their jobs. In the case of cross domain system accreditors, however, this precaution can backfire in interesting ways; certain undesirable information flows are forced, and certain desirable information flows are inhibited.

If, however, all accreditors are simply cleared to the highest security level, no information leakage occurs. Relaxing the security rules paradoxically improves security.

The certifier behaviour model is less well developed but offers a means whereby the cross domain solution developer can exert a measure of control over the schedule, and hence the cost of certification testing, in a specific example of the more general observation that some evidence exists to suggest a new measure of organisational maturity for software development organisations that can predict the success or failure of security certification activities.

7.2 Future Work

Consider the problems of simulating a market amongst accreditors who are constrained from communicating freely about their individual assessments of residual risk of a cross domain system because of security classification rules. It is a weird sort of market in which participants offer to buy and sell commodities that they do not know the value of, although someone else does. Since some of the accreditors are prohibited from describing the exact details of a threat or a risk mitigation, or even the lack of any known risk mitigation for a threat with no countermeasure, they must in some way signal (in Spence's use of the word) the actual value of the residual risk as they perceive it. From the response received at a recent conference when the

idea was mentioned, the researcher is not the only one thinking along the lines of a market for risk.

A new tool is being developed, called *nihil obstat*, that is intended to facilitate the determination of an equilibrium in the market for residual risk amongst cross domain system accreditors by soliciting a series of bid/ask quotations from accreditors at different security classification levels and using them to set a 'market price' for the residual risk that each accreditor is prepared and willing to accept.

Further applications to incident response and continuous monitoring may be found; although the time scale is shorter, the required communication between executives and the incident officer, through an incident manager to incident responders, resembles the structure of a C&A activity and analogous information flows may be expected there.

7.3 Conclusion

Oftentimes failure investigation reports eventually are used for political or litigation purposes; moreso than any other entity, the Marine Accident Investigation Board (UK) has the right idea; its reports contain the following caveat on the first page:

> 'This report is not written with litigation in mind and, pursuant to Regulation 13(9) of the Merchant Shipping (Accident Reporting and Investigation) Regulations 2005, shall be inadmissible in any judicial proceedings whose purpose, or one of whose purposes is to attribute or apportion liability or blame' [148].

It is hoped that this thesis and the follow-on journal paper and conference presentation detailing the full story of the successful R'' certification activity—the first ever under the new ICD 503 rules—will bring a wider audience to the fascinating problem of cross domain C&A, and when the expected

flood of Electronic Health Record (EHR) interconnections arrives in the near future, that the lessons learnt by the classified world will benefit the 'unclas' side as well.

Glossary of Specialised Terms

acceptance test

Testing performed by the developer, which may be witnessed by the certifier, to ensure that a cross domain solution satisfies a specific set of requirements for functionality. Synonymous with Factory [or Formal] Acceptance Test (FAT) or alpha testing in some certification and accreditation processes.

accreditor

A person, usually a government official, with the authority to accept responsibility for the correct operation of a cross domain system. Synonymous with Designated Approving Authority (DAA) or Designated Accrediting Authority (DAA).

accreditation

Formal acceptance by an accreditor of the residual risk for operation of a particular system in a particular environment. Synonymous with Security Test and Evaluation (ST&E) or beta II testing [74, 108].

certification

Statement of approval by an authority, based on testing and examina-

tion of evidence, that a cross domain solution meets specified security functional requirements to a specified level of assurance. Synonymous with Certification Test and Evaluation (CT&E) or beta I testing [74, 108].

certifier

Synonymous with certification authority.

Compliance Assessment Team (DNI CAT)

Director of National Intelligence (DNI) Compliance Assessment Team (CAT)

Controlled Interface (CI)

See *cross domain solution* or *cross domain system.* The term 'controlled interface' originated with DCID 6/3 [74].

Cross Domain Solution (CDS)

A cross domain solution is a product, usually comprising certified software running on approved hardware, intended for the purpose of interconnecting two or more communicating endpoints at different security levels. The functionality of a cross domain solution may include one-way, two-way, or multi-way information flows, automatic sanitisation, upgrading, downgrading, and guarding.

Cross Domain System (CDS)

A cross domain system comprises a cross domain solution and two or more communicating endpoints.

data owner

In the context of a cross domain system, the agency having responsibility

for security of information within a particular security domain, often represented directly by an accreditor.

A data owner cares not about information in other data owners' security domains, except to the extent that it might be infested with malware, the other side of the cross domain interface coin.

Designated Approving Authority (DAA)

see accreditor.

developer

The software writer, or sometimes hardware builder or systems integrator, of a cross domain solution.

downgrade

To change the security classification level of some information from a higher to a lower security level, for example from SECRET to unclassified. Downgrading is almost always done by sanitisation.

Independent Verification and Validation (IV&V)

Post–acceptance-test re-testing of all acceptance test cases by someone other than the developer. When performed at the behest of a certifier, synonymous with ' regression testing' in some certification and accreditation processes.

installer

Developer's representative in the field who sets up and configures a cross domain solution for operation in a cross domain system.

penetration testing

Testing performed by or at the behest of a certifier with the aim of discovering security weaknesses in the system under test, rather than

verifying functionality or performance. It is formally performed in the Beta II phase.

redaction

The act of removing or obscuring information from some data for the purpose of protecting sensitive information in the original. The redacted data may be able to be classified at a lower security level, but the reclassification is actually done by a downgrade.

sanitisation

The act of changing the content or presentation of some data to remove information, degrade its precision, obscure the source of the information, or make it appear to have originated from a different source, or in a different format. Sanitisation includes redaction.

upgrade

To change the security classification level of some information from a lower to a higher security level, for example from unclassified to secret.

Bibliography

[1] George A. Akerlof. The market for 'lemons': Quality uncertainty and the market mechanism. *Quarterly Journal of Economics*, 84(3):488–500, August 1970.

[2] Richard J. Aldrich. *GCHQ*. Harper, London, 2010.

[3] Jim Alves-Foss, Paul W. Oman, Carol Taylor, and W. Scott Harrison. The MILS architecture for high-assurance embedded systems. *Int. J. Emb. Sys.*, 2(3/4):239–247, 2006.

[4] Ross J. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley Computer Publishing, New York, 2001.

[5] Ross J. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. John Wiley & Sons, Inc., Indianapolis, Indiana, second edition, 2008.

[6] Christopher Andrew. *For the President's Eyes Only: Secret Intelligence and the American Presidency from Washington to Bush*. Harper Collins, New York, 1st edition, 1995.

[7] Antoine Arnauld. *La logique, ou, L'art de penser*. Monastery of Port Royal, 1662.

[8] Assistant Secretary of Defence (Command, Control, Communications, and Intelligence) [ASD(C3I)]. *Directive 8500.1, Subject: Information Assurance*. United States Department of Defence, October 24, 2002.

[9] Assistant Secretary of Defence (Command, Control, Communications, and Intelligence) [ASD(C3I)]. *Instruction 8500.2, Subject: Information Assurance (IA) Implementation*. United States Department of Defence, February 6, 2003.

[10] Assistant Secretary of Defence for Networks and Information Integration/Department of Defence Chief Information Officer [ASD(NII)/DoD CIO]. *Directive 8500.01E, Subject: Information*

Assurance (IA). United States Department of Defence, October 24, 2002. (Certified current as of April 23, 2007).

[11] Norman R. Augustine. *Augustine's Laws*. American Institute of Aeronautics and Astronautics, Reston, Virginia, 6th edition, 1997.

[12] James Backhouse and Gurpreet Dhillon. Structures of responsibility and security of information systems. *European Journal of Information Systems*, 5(1):2–9, 1996.

[13] Marianne Bailey. The Unified Cross Domain Management Office: Bridging security domains and cultures. *Crosstalk: The Journal of Defence Software Engineering*, pages 21–23, July 2008.

[14] James Bamford. *The Puzzle Palace: Inside the National Security Agency, America's Most Secret Intelligence Organization*. Penguin Books, New York, 1983.

[15] James Bamford. *Body of Secrets*. Anchor, New York, 2002.

[16] James Bamford. *The Shadow Factory*. Doubleday, New York, 2008.

[17] Robert Barrass. *Scientists Must Write*. Routledge Falmer, 2nd edition, 2002.

[18] Richard Baskerville. The developmental duality of information systems security. *Journal of Management Systems*, 4(1):1–12, 1992.

[19] D. Elliott Bell and Leonard J. LaPadula. Secure computer systems: Mathematical foundations. Technical Report 2547, Volume I, the MITRE Corporation, March 1973.

[20] David Elliott Bell. Looking back at the Bell–LaPadula model. In *21st Annual Computer Security Applications Conference*, pages 337–351, Tucson, Arizona, USA, 5–9 December 2005.

[21] D.E. Bell and L.J. LaPadula. Secure computer system: Unified exposition and MULTICS interpretation. Technical Report 2997, the MITRE Corporation, March 1976.

[22] Emma Bell. Managerialism and management research: Would Melville Dalton get a job today? In Bill Lee and Catherine Cassell, editors, *Challenges and Controversies in Management Research*, pages 122–137. Routledge, New York, 2011.

[23] Steven M. Bellovin. Frank Miller: Inventor of the one-time pad. *Cryptologia*, 35(3):203–222, 2011.

[24] Peter L. Bernstein. *Against the Gods*. John Wiley & Sons, New York, 1996.

[25] Kenneth J. Biba. Integrity considerations for secure computer systems. Technical Report MTR-3153, the MITRE Corporation, April 1977.

[26] Eric Bidstrup. Common Criteria and answering the question 'is it safe?'. The *Security Development Lifecycle* blog, Thursday, December 20, 2007. URL: ⟨`http://blogs.msdn.com/b/sdl/archive/2007/12/20/common-criteria-and-answering-the-question-is-it-safe.aspx`⟩.

[27] Richard Biernacki. *Reinventing Evidence in Social Inquiry*. Palgrave Macmillan, New York, first edition, July 2012.

[28] John Black, Nigar Hashimzade, and Gareth Myles, editors. *Oxford Dictionary of Economics*. Oxford University Press, Oxford, third edition, 2009.

[29] Trevor Blackwell. The `analogtv.c` source code. in *`XScreenSaver`: a collection of free screen savers for X11 and MacOS*, by Jamie Zawinski *et al.*, 2003. URL: ⟨`http://www.jwz.org/xscreensaver/xscreensaver-5.22.tar.gz`⟩.

[30] Rainer Böhme and Márk Félegyházi. Optimal information security investment with penetration testing. In *Decision and Game Theory for Security*, volume LNCS 6442. Springer, Berlin, Germany, November 22–23, 2010.

[31] Rainer Böhme and Tyler Moore. The iterated weakest link: A model of adaptive security investment. In *Workshop on the Economics of Information Security (WEIS)*, London, 24–25 June 2009.

[32] David L. Boslaugh. *When Computers Went to Sea: The Digitization of the United States Navy*. IEEE Computer Society Press, Los Alamitos, California, 1999.

[33] George E.P. Box. Robustness in the strategy of scientific model building. In Robert L. Launer and Graham N. Wilkinson, editors, *Robustness in Statistics*, pages 201–236. Academic Press, 1979.

[34] George E.P. Box and Norman R. Draper. *Empirical Model-Building and Response Surfaces*. Wiley series in probability and mathematical statistics. John Wiley & Sons, Oxford, England, 1987.

[35] G.E.P. Box. Robustness in the strategy of scientific model building. Technical report, University of Wisconsin—Madison Mathematics Research Centre, May 1979. DTIC.mil accession number ADA070213, ordered from NTIS on 7th March 2012 for $33.00.

[36] Fredric Boyce and Douglas Everett. *SOE: The Scientific Secrets*. Sutton Publishing Ltd, Stroud, Gloucestershire, 2004.

[37] Fred Brandes. John Wilkins—Mercury the Secret & Swift Messenger. *Cryptologic Nooks & Crannies*, September 24, 2007. URL: ⟨`http://www.codasaurus.com/ubb/Forum5/HTML/000258.html`⟩.

[38] Bob Briscoe, Andrew Odlyzko, and Benjamin Tilly. Metcalfe's law is wrong. *IEEE Spectrum*, 43(7):34–39, July 2006.

[39] Fred P. Brooks. No silver bullet: Essence and accident in software engineering. In H.-J. Kugler, editor, *Information Processing 1986, the Proceedings of the IFIP Tenth World Computing Conference*, pages 1069–1076, 1986.

[40] Frederick P. Brooks. *The Mythical Man-Month*. Addison–Wesley Professional, 2nd edition, 1995.

[41] Louis Brown. *A Radar History of World War II: Technical and Military Imperatives*. Institute of Physics Publishing, Bristol, 1999.

[42] Antony Bryant and Kathy Charmaz, editors. *The SAGE Handbook of Grounded Theory*. SAGE Publications Ltd, Los Angeles, California, 2010.

[43] Robert Buderi. *The Invention that Changed the World: How a Small Group of Radar Pioneers Won the Second World War and Launched a Technological Revolution*. Touchstone, New York, 1998.

[44] Stephen Budianski. *Her Majesty's Spymaster: Elizabeth I, Sir Francis Walsingham, and the Birth of Modern Espionage*. Penguin Books, 80 Strand, London WC2R 0RL, England, 2006.

[45] James G. Burton. *The Pentagon Wars: Reformers Challenge the Old Guard*. Naval Institute Press, Annapolis, Maryland, 1993.

[46] Roger Caslow. Talk on ICD 503, the replacement for DCID 6/3 in the intelligence community. In *Unified Cross Domain Management Office (UCDMO) Conference*, Chicago, Illinois, 1–4 August 2011.

[47] Maciej Ceglowski. A rocket to nowhere. *Idle Words* blog, August 2005. URL: ⟨`www.idlewords.com/2005/08/a_rocket_to_nowhere.htm`⟩.

[48] Centre for Cryptologic History. *The Rare Book Collection*. National Security Agency, Fort Meade, Maryland, 2008.

[49] Kathy Charmaz. *Constructing Grounded Theory: A Practical Guide through Qualitative Analysis*. Sage Publications Ltd, London, first edition, 2006.

[50] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexi Czeskis, Franziska Roesner, and Tadayohi Kohno. Comprehensive experimental analyses of automotive attack surfaces. In *20th USENIX Security Symposium*, San Francisco, California, August 8–12, 2011.

[51] William R. Cheswick and Steven M. Bellovin. *Firewalls and Internet Security: Repelling the Wily Hacker*. Addison–Wesley, Reading, Massachusetts, first edition, 1994.

[52] John D. Clark. *Ignition! An Informal History of Liquid Rocket Propellants*. Rutgers University Press, New Brunswick, New Jersey, 1972.

[53] H.M. Collins. *Changing Order: Replication and Induction in Scientific Practice*. University of Chicago Press, New Ed edition, 1992.

[54] H.M. Collins and R.G. Harrison. Building a TEA laser: The caprices of communication. *Social Studies of Science*, 5, November 1975.

[55] David Colquhoun. An investigation of the false discovery rate and the misinterpretation of *p*-values. *Royal Society Open Science*, 19th November 2014.

[56] Committee on National Security Systems. *National Policy Governing the Acquisition of Information Assurance (IA) and IA-Enabled Information Technology (IT) Products*. U.S. National Security Telecommunications and Information Systems Security Policy (NSTISSP) No. 11, June 2003.

[57] Common Criteria Sponsoring Organizations. *Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance requirements*. Common Criteria Sponsoring Organisations, August 2005. Version 2.3, CCMB-2005-08-003.

[58] Common Criteria Sponsoring Organizations. *Common Criteria for Information Technology Security Evaluation*. Common Criteria Sponsoring Organisations, September 2006. Version 3.1, Revision 1, three volumes, CCMB-2006-09-001.

[59] Common Criteria Sponsoring Organizations. *Common Methodology for Information Technology Security Evaluation: Evaluation Methodology*. Common Criteria Sponsoring Organisations, September 2007. Version 3.1, Revision 2, CCMB-2007-09-004.

[60] Jennet Conant. *Tuxedo Park: A Wall Street Tycoon and the Secret Palace of Science that Changed the Course of World War II*. Simon & Schuster, New York, 2002.

[61] Contactless and Mobile Payments Council. What makes a smart card secure? White Paper CPMC-08002, Smart Card Alliance, October 2008.

[62] Elliott Vanveltner Converse. *Rearming for the Cold War, 1945–1960*, volume 1 of *History of acquisition in the Department of Defense*. Office of the Secretary of Defense, Historical Office, Washington, D.C., 2012. PDF.

[63] Melvin E. Conway. How do committees invent? *Datamation*, 14(4):28–31, April 1968.

[64] Melvin E. Conway. The Tower of Babel and the fighter plane (keynote address). In *3rd International Workshop on Replication in Empirical Software Engineering Research (RESER2013)*, Baltimore, Maryland, 9th October 2013. IEEE.

[65] Jack Copeland, editor. *Colossus: The Secrets of Bletchley Park's Code-breaking Computers*. Oxford University Press, Oxford, 2006.

[66] B. Corby. IBM 2750 voice and data switching system: Organization and functions. *IBM Journal of Research and Development*, 13(4):408–415, July 1969.

[67] Chris Craddock. Tales from the "academic mafia" — the REAL inside story of Westpac's CS90. *The Software Practitioner*, 9(2):1, 4–6, March–April 1999.

[68] Everett U. Crosby. Fire prevention. *Annals of the American Academy of Political and Social Science*, 26:224–238, 1905.

[69] Richard L. Daft. Why i recommended that your manuscript be rejected and what you can do about it. In L.L. Cummings and Peter J. Frost, editors, *Publishing in the Organizational Sciences*, pages 164–182. Sage Publications, Inc., Thousand Oaks, California, 2nd edition, 1995.

[70] Melville Dalton. *Men Who Manage: Fusions of Feeling and Theory in Administration*. John Wiley & Sons, Inc., New York, 1959.

[71] Melville Dalton. Preconceptions and methods in *Men Who Manage*. In Phillip E. Hammond, editor, *Sociologists at Work: Essays on the Craft of Sociological Research*, pages 50–95. Basic Books, New York, 1964.

[72] Blaise de Vigenère. *Traicté des Chiffres; ou, Secrètes Manières d'escrire*. Abel L'Angelier, Paris, 1585.

[73] Andrew Dequasie. *The Green Flame: Surviving Government Secrecy*. American Chemical Society, Washington, D.C., 1991.

[74] Director of Central Intelligence. *Protecting Sensitive Compartmented Information within Information Systems (DCID 6/3)—Manual.* United States Government, 1 August 2000.

[75] Jeffrey E. Duven. Special conditions: The Boeing Company, Models 737-700, -700C, -800, -900ER, -7, -8, and -9 Series airplanes; airplane electronic systems security protection from unauthorized external access. *Federal Register*, 79(109):32640–1, 6th June 2014. Federal Aviation Administration Rule 14 CFR Part 25 Special Conditions No. 25-550-SC.

[76] John Edwards. Cross-domain office aims for secure data sharing. *Defense Systems*, 28th July 2011. ⟨`http://defensesystems.com/articles/2011/07/18/defense-it-2-cross-domain-sharing.aspx?s=ds_030811&admgarea=TC_DEFENSE`⟩.

[77] Kurt Elean. Talk on the DoD IA policy portfolio managed by DoD CIO, DASD(NII). In *Unified Cross Domain Management Office (UCDMO) Conference*, Chicago, Illinois, 1–4 August 2011.

[78] Electronic Frontier Foundation. *Cracking DES: Secrets of Encryption Research, Wiretap Politics and Chip Design.* O'Reilly & Associates, Sebastopol, California, 1998.

[79] *Bundessamt für Sicherheit in der Informationstechnik Deutsches IT-Sicherheitzertifikat.* Certification report PR/SM LPAR for the IBM eServer zSeries z890 and z990 from International Business Machines Corporation (IBM). Certification Report BSI-DSZ-CC-0279-2005, BSI, 13th May 2005.

[80] *Sanctissimi D.N. Pius XII, iussu editus*, editor. *Index Librorum Prohibitorum. Typis Polyglottis Vaticanis*, *Romæ*, 1948.

[81] Thomas Ernst. The numerical-astrological ciphers in the third book of Trithemius's *Steganographia. Cryptologia*, XXII(4):318–341, October 1998.

[82] Shamal Faily. *A framework for usable and secure system design.* PhD thesis, University of Oxford, Hilary Term 2011.

[83] John Falconer. Cryptomenyſis Patefacta*: Or The Art of Secret Information Diſcloſed Without a KEY. Containing, Plain and Demonſtrative Rules, for Decyphering All Manner of SECRET WRITING. With exact methods, for Reſolving Secret Intimations by SIGNS or GESTURES, or in SPEECH. As alſo an inquiry into the Secret ways of CONVEYING written meſſages, and the ſeveral MYSTERIOUS*

PROPOSALS for Secret Information, mentioned by Trithemius*, &c.* Daniel Brown, London, 1685.

[84] Christopher Felix. *A Short Course in the Secret War.* Madison Books, Lanham, Maryland, 2000.

[85] Neils Ferguson and Bruce Schneier. *Practical Cryptography.* Wiley Publishing, Inc., Indianapolis, Indiana, 2003.

[86] A. Finkelstein and J. Dowell. A comedy of errors: the London Ambulance Service case study. In *Proceedings of the 8th International Workshop on Software Specification and Design*, pages 2–4, Schloss Velen, Germany, March 22–23, 1996.

[87] Anthony Finkelstein. Report of the inquiry into the London ambulance service, February 1993.

[88] Ivan Fléchais. *Designing Secure and Usable Systems.* PhD thesis, University College, London, 2005.

[89] Eric Fleischman. Handbook for networked local area networks in aircraft. Final Report DOT/FAA/AR-08/35, U.S. Department of Transportation, Federal Aviation Administration, 2008.

[90] Michael T. Flynn. Memorandum for deputy commanding general for support, USFOR-A; subject: (U) advanced analytical capability joint urgent operational need statement. USFOR-J2, 2nd July 2010.

[91] Greg Freiherr. Orbiter autopsies. *Air & Space*, 27(1):24–29, April/May 2012.

[92] William Friedman. *Military Cryptanalysis, Part I.* Aegean Park Press, Walnut Creek, California, 1996.

[93] William A. Gamson and Norman A. Scotch. Scapegoating in baseball. *American Journal of Sociology*, 70(1):69–72, July 1964.

[94] Harold Garfinkel. *Studies in Ethnomethodology.* Prentice-Hall, Englewood Cliffs, New Jersey, 1967.

[95] Bartek Gedrojc. Fort Fox hardware data diode security target: Common Criteria FFHDD—EAL4+, August 21, 2009. Version 1.06.

[96] Bartek Gedrojc. Fort Fox hardware data diode security target: Common Criteria FFHDD—EAL7+, June 3, 2010. TD-10-02-00103, Version 2.04.

[97] Genesis Mishap Investigation Board. Genesis mishap investigation board (MIB) report, volume I. Technical report, National Aeronautics and Space Administration, 2005.

[98] Malcolm Gladwell. Trust no one: Kim Philby and the hazards of mistrust. *The New Yorker*, XC(21):70–5, July 28th 2014.

[99] Barney G. Glaser and Anselm L. Strauss. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Transaction, New Brunswick, New Jersey, USA, 1967.

[100] Robert L. Glass. *Software Runaways*. Prentice Hall, 1st edition, 1997.

[101] Robert L. Glass. Buzzwordism and the epic $150 million software debacle. *Comm. ACM*, 42(8):17–19, August 1999.

[102] Robert L. Glass. *Computing Calamities*. Prentice-Hall PTR, Upper Saddle River, New Jersey, 1999.

[103] Francis Godwin. *Nuncius Inanimatus*. Unknown, in Utopia, 1629.

[104] Green Hills Software, Inc. Safety critical products: INTEGRITY-178B RTOS. URL: ⟨`http://www.ghs.com/products/safety_critical/integrity-do-178b.html`⟩, 2008.

[105] Hugh Gusterson. A pedagogy of diminishing returns: Scientific involution across three generations of nuclear weapons science. In David Kaiser, editor, *Pedagogy and the Practice of Science: Historical and Contemporary Perspectives*. The MIT Press, 2005.

[106] Ian Hacking. *The Emergence of Probability: A Philosophical Study of Early Ideas about Probability, Induction, and Statistical Inference*. Cambridge University Press, London, 1972.

[107] Katie Hafner. *Where Wizards Stay Up Late: The Origins of the Internet*. Simon & Schuster Ltd, New York, 1996.

[108] Susan Hansche. *The Official $(ISC)^2$ Study Guide for the Information System Security Engineering Professional (ISSEP) Exam*. Taylor & Francis Ltd, Boca Raton, Florida, 2005.

[109] David M. Harland and Ralph D. Lorenz. *Space Systems Failures: Disasters and Rescues of Satellites, Rockets and Space Probes*. Praxis, Chichester, UK, 2005.

[110] F.G. Heath. Origins of the binary code. *Scientific American*, 227(2):76–83, August 1972.

[111] John L. Hennessy and David A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kauffmann Publishers, San Mateo, California, 1990.

[112] Wesley H. Higaki. A proposal for a COTS assurance package. In *Proceedings of the 9th International Common Criteria Conference (ICCC)*, Jeju, Korea, 23–25 September 2008. URL: ⟨`http://www.9iccc.kr/program/pdf/A2406.pdf`⟩.

[113] Steve Hill and Duncan Harris. Time flies: Examining timeliness in ALC_FLR.3. In *Proceedings of the 9th International Common Criteria Conference (ICCC)*, Jeju, Korea, 23–25 September 2008. URL: ⟨`http://www.9iccc.kr/program/pdf/B2307.pdf`⟩.

[114] Jim Hodges. The price of security. *C4ISR Journal*, 9(8):20–22, September 2010.

[115] R.T.M. Huisman. Fort Fox hardware data diode, version FFHDD2+. Certification Report NSCIB-CC-09-11025, TNO Certification, June 16, 2010. URL: ⟨`http://www.commoncriteriaportal.org/files/epfiles/Certification Report NSCIB-CC-09-11025-CR2.pdf`⟩.

[116] Robert Hutchinson. *Elizabeth's Spymaster: Francis Walsingham and the Secret War that Saved England*. Thomas Dunne Books, New York, 2007.

[117] IBM Corporation. *Vocabulary for Data Processing, Telecommunications, and Office Systems*. IBM Corporation, Poughkeepie, New York, seventh edition, July 1981.

[118] IBM Corporation. Public version of the security target for PR/SM for the IBM eServer zSeries 900 EAL5 certification, December 5, 2002. Version 1.22.

[119] Information Sciences Institute. Transmission control protocol. Request for Comment (RFC) 793, University of Southern California, September 1981.

[120] International Information Systems Security Certification Consortium [$(ISC)^2$]. Member counts, July 2012.

[121] William Jackson. Under attack: Common Criteria has loads of critics, but is it getting a bum rap? *Government Computer News*, 26(21), August 13, 2007.

[122] Roger G. Johnston. Security maxims, July 2, 2009. Retrieved 13th April 2010 from ⟨`www.ne.anl.gov/capabilities/vat/seals/maxims.html`⟩.

[123] R.V. Jones. *Most Secret War: British Scientific Intelligence 1939–1945*. Hamish Hamilton, London, 1978.

[124] David Kahn. *The Codebreakers*. Scribner, New York, 1996.

[125] Jan Kallberg. Common criteria meets realpolitik: Trust, alliances, and potential betrayal. *IEEE Security & Privacy*, 10(4), July/August 2012.

[126] Paul A. Karger and Roger R. Schell. Multics security evaluation: Vulnerability analysis. Technical Report ESD-TR-74-193, Vol. II, United States Air Force, Information Systems Technology Applications Office, Deputy for Command and Management Systems, Electronic Systems Division (AFSC), June 1974.

[127] Paul A. Karger and Roger R. Schell. Thirty years later: Lessons from the Multics security evaluation. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, Las Vegas, Nevada, USA, December 9–13, 2002.

[128] T.A. Kletz. *Lessons from Disaster—How Organisations Have No Memory and Accidents Recur*. Institution of Chemical Engineers, Rugby, UK, 1993.

[129] Trevor Kletz. *An Engineer's View of Human Error*. Institution of Chemical Engineers, Rugby, Warwickshire CV21 3HQ, UK, second edition, 1991.

[130] Trevor Kletz. *Still Going Wrong!: Case Histories of Process Plant Disasters and How They Could Have Been Avoided*. Gulf Professional Publishing, Burlington, Massachusetts, 2003.

[131] Trevor Kletz and Paul Amyotte. *Process Plants: A Handbook for Inherently Safer Design*. CRC Press, Boca Raton, Florida, second edition, 2010.

[132] Trevor A. Kletz. *HAZOP & HAZAN: Identifying and Assessing Process Industry Hazards*. Institution of Chemical Engineers, Rugby, Warwickshire CV21 3HQ, UK, fourth edition, 1999.

[133] Trevor A. Kletz. *What Went Wrong?: Case Histories of Process Plant Disasters*. Elsevier, Burlington, Massachusetts, fourth edition, 1999.

[134] Jukka Korpela. How all human communication fails, except by accident, or a commentary of Wiio's laws. *IT and Communication*, 2010.

[135] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage. Experimental security analysis of a modern automobile. In *IEEE Symposium on Security & Privacy*, Oakland, California, 16–19 May 2010.

[136] Rick Lehtinen, Deborah Russell, and G.T. Gangemi, Sr. *Computer Security Basics*. O'Reilly & Associates, Sebastopol, California, second edition, 2006.

[137] Timothy E. Levin, Cynthia E. Irvine, Clark Weissman, and Thuy D. Nguyen. Analysis of three multilevel security architectures. In *Proceedings of the 2007 ACM workshop on Computer security architecture*, pages 37–46, Fairfax, Virginia, USA, 2nd November 2007.

[138] Steven Levy. *Hackers: Heroes of the Computer Revolution*. Penguin Books, New York, 1984.

[139] James Littell. *The Company*. Macmillan, New York, 2002.

[140] Lockheed Martin Corporation. Radiant Trust Gold (RTG) 1.0 security target. Document SSYR-050-10424-000:66, 2006.

[141] Lockheed Martin Corporation. Radiant Mercury 5.0 capabilities, 27th May 2011.

[142] Joe Loughry. Unsteady ground: Certification to unstable criteria. In *Proc. Second International Conference on Advances in System Testing and Validation Lifecycle*, Nice, France, 22–27 August 2010.

[143] Joe Loughry. Information asymmetry in classified cross domain system security accreditation. In *Computer and Electronics Security Applications Rendez-vous (C&ESAR) 2012*, pages 19–28, Rennes, France, 20–22 November 2012.

[144] Joe Loughry. A model of certifier and accreditor risk calculation for multi-level systems. In *13th IEEE Conference on Technologies for Homeland Security (HST'13)*, Boston, Massachusetts, 12–14 November 2013.

[145] Paul Luff. Ethnographic methods: an approach using video-based fieldwork. Part of the *Requirements Engineering* (REN) course, University of Oxford Software Engineering Programme, June 2009.

[146] David E. Lundstrom. *A Few Good Men From Univac*. The MIT Press, Cambridge, Massachusetts, 1987.

[147] M. Mangel and F.J. Samaniego. Abraham Wald's work on aircraft survivability. *Journal of the American Statistical Association*, 79:259–267, 1984.

[148] Marine Accident Investigation Branch. Report on the investigation of the catastrophic failure of a capacitor in the aft harmonic filter room on board RMS *Queen Mary 2* while approaching Barcelona 23

September 2010. Accident Report 28/2011, Marine Accident Investigation Branch, Mountbatten House, Grosvenor Square, Southampton, United Kingdom, SO15 2JU, December 2011.

[149] Andrew P. Martin. Personal communication, 15th July 2011.

[150] Gerald M. McCue. IBM's Santa Teresa Laboratory—architectural design for program development. *IBM Systems J.*, 17(1):4–25, 1978.

[151] John Mills. *The Engineer in Society*. D. Van Nostrand Co. Inc., New York, 1946. DU Penrose Library, TA 157.M5 books lower level, stamped 'Mary Reed Library, University of Denver'.

[152] Harriet B. Moore and Sidney J. Levy. Artful contrivers: A study of engineers. *Personnel*, 28:148–153, 1951.

[153] Steven J. Murdoch, Mike Bond, and Ross Anderson. How certification systems fail: Lessons from the Ware report. *IEEE Security & Privacy*, 10(6):40–44, November/December 2012.

[154] National Aeronautics and Space Administration. *NOAA N-Prime* mishap investigation. Final report, Mishap Investigation Board, September 13, 2004.

[155] National Audit Office. The failure of the FiReControl project, 1st July 2011. Report by the Comptroller and Auditor General, HC 1272, Session 2010–2012.

[156] National Computer Security Centre. *Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria*, 31 July 1987. NCSC-TG-005 Version-1.

[157] National Computer Security Centre. *A Guide to Understanding Covert Channel Analysis of Trusted Systems*, November 1993. NCSC-TG-030 Version 1.

[158] National Information Assurance Partnership. Common Criteria certificate, 9 April 2003. PC Guardian, EAL1.

[159] National Information Security Program. *Operating Manual*, February 28, 2006. DoD 5220.22-M.

[160] National Institute of Standards and Technology. *FIPS PUB 140-2: Security Requirements for Cryptographic Modules*, May 25, 2001. Federal Information Processing Standards Publication (Supercedes FIPS PUB 140-1, 1994 January 11).

[161] National Institute of Standards and Technology. *Guide for Applying the Risk Management Framework to Federal Information Systems*, February 2010. NIST Special Publication 800-37 Revision 1.

[162] National Institute of Standards and Technology. *Managing Information Security Risk: Organization, Mission, and Information System View*, March 2011. NIST Special Publication 800-39.

[163] National Reconnaissance Office. *Certifying & Accrediting NRO Information Systems: Implementers' Course*, fifth edition, unknown year. Unclassified//For Official Use Only.

[164] National Security Agency, Central Security Service. Fact sheet: NSA Suite B cryptography, 2008. URL: ⟨`http://www.nsa.gov/ia/Industry/crypto_suite_b.cfm`⟩.

[165] NIAP CCEVS. New *NIAP CCEVS* strategy for FY10, March 16, 2009.

[166] John David North, Arjo Vanderjagt, and Lodi Nauta, editors. *Between Demonstration and Imagination: Essays in the History of Science and Philosophy*. Brill Academic Publishers, 1999.

[167] Andrew Odlyzko. The forgotten discovery of gravity models and the inefficiency of early railway networks. Preliminary version, September 1, 2014, preprint.

[168] Office of Inspector General. Faster, better, cheaper: Policy, strategic planning, and human resource alignment. Audit Report IG-01-009, National Aeronautics and Space Administration, March 13, 2001.

[169] Office of the Director of National Intelligence. Classified information nondisclosure agreement, 2013. Standard Form 312 (Rev 7-2013), NSN 7450-01-280-5499.

[170] Office of the Intelligence Community Chief Information Officer (IC CIO). Intelligence community inter-domain transfer policy, 9 November 1999. Unclassified//For Official Use Only, Draft—For Review.

[171] James M. Olson. The ten commandments of counterintelligence. *Studies in Intelligence*, 11, Fall/Winter 2001.

[172] Severo M. Ornstein. *Computing in the Middle Ages: A View From the Trenches 1955–1983*. 1st Book Publishing, Miami, Florida, 2002.

[173] Donn B. Parker. *Crime by Computer*. Charles Scribner's Sons, New York, 1st edition, 1976.

[174] Stephen K. Parker and Martin Skitmore. Project management turnover: causes and effects on project performance. *International Journal of Project Management*, 23(3):205–214, April 2005.

[175] David A. Patterson and John L. Hennessy. *Computer Organization & Design: The Hardware/Software Interface*. Morgan Kauffmann Publishers, San Mateo, California, 1994.

[176] Sir David Pepper. From the Cold War to the digital age: the transformation of GCHQ, 20th October 2008. a lecture given to the Oxford Intelligence Group, in the Large lecture room at Nuffield College, Oxford.

[177] Jeffrey Pfeffer. Four laws of organizational research. In Andrew H. Van de Ven and William F. Joyce, editors, *Perspectives on Organization Design and Behavior*, pages 409–418. John Wiley & Sons, New York, 1981.

[178] William Poole. *Nuncius Inanimatus*—seventeenth century telegraphy: the schemes of Francis Godwin and Henry Reynolds. *The Seventeenth Century*, XXI(1):45–72, Spring 2006.

[179] Karl R. Popper. *The Logic of Scientific Discovery*. Routledge, London, 2002.

[180] President of the United States. Executive order 13526: Classified national security information, January 5, 2010.

[181] Thomas Ptacek. What Common Criteria certification means. *Matasano Security* blog, June 19th, 2006. URL: ⟨`http://www.matasano.com/log/331/what-common-criteria-certification-means/`⟩.

[182] Emerson W. Pugh. *Memories That Shaped an Industry: Decisions Leading to IBM System/360*. MIT Press, Cambridge, Massachusetts, 1984.

[183] Emerson W. Pugh, Lyle R. Johnson, and John H. Palmer. *IBM's 360 and Early 370 Systems*. MIT Press, Cambridge, Massachusetts, 1991.

[184] Nithya Rachamadugu. Scoping the TOE. In *Proceedings of the 9th International Common Criteria Conference (ICCC)*, Jeju, Korea, 23–25 September 2008. URL: ⟨`http://www.9iccc.kr/program/pdf/C2401.pdf`⟩.

[185] Eric S. Raymond. *The New Hacker's Dictionary*. The MIT Press, Cambridge, Massachusetts, 1993.

[186] [Redacted]. Eurocrypt '92 reviewed. *Cryptolog*, XX(1):12–19, 1994.

[187] Kent C. Redmond and Thomas M. Smith. *From Whirlwind to MITRE: The R&D Story of The SAGE Air Defence Computer*. MIT Press, Cambridge, Massachusetts, 2000.

[188] Thomas Reed. *At the Abyss: An Insider's History of the Cold War*. Presidio Press, New York, 2005.

[189] Jim Reeds. Solved: The ciphers in Book III of Trithemius's *Steganographia*. *Cryptologia*, 22(4):291–319, October 1998.

[190] Tommaso Agostino Ricchini, editor. *Index librorum prohibitorum sanctissimi D.N. Benedicti XIV pontificis maximi jvssv recognitus, atque editus*. ex typographia reverendæ Cameræ apostolicæ, Romæ, 1758.

[191] Jeffrey T. Richelson. *The Wizards of Langley: Inside the CIA's Directorate of Science and Technology*. Westview Press, Boulder, Colorado, USA, 2002.

[192] Jeffrey T. Richelson. *The US Intelligence Community*. Westview Press, 2465 Central Avenue, Boulder, Colorado 80301-2877 USA, 5th edition, 2007.

[193] Ron Ross, Arnold Johnson, Stu Katzke, Patricia Toth, Gary Stoneburner, and George Rogers. *Guide for Assessing the Security Controls in Federal Information Systems*, July 2008. NIST Special Publication 800-53A.

[194] Johann Rost and Robert L. Glass. Subversion and lying: The dark side of IT politics. *Cutter IT Journal*, 18(4), April 2005.

[195] Johann Rost and Robert L. Glass. *The Dark Side of Software Engineering: Evil on Computing Projects*. IEEE Computer Society Press, Los Alamitos, California, 2011.

[196] Frank B. Rowlett. *The Story of Magic, Memoirs of an American Cryptologic Pioneer*. Aegean Park Press, Walnut Creek, California, 1998.

[197] RTCA, Inc. *Software Considerations in Airborne Systems and Equipment Certification*, December 1, 1992. DO-178B.

[198] RTCA, Incorporated. *DO-178C Software Considerations in Airborne Systems and Equipment Certification*, December 2011.

[199] Debby Russell and G.T. Gangemi, Sr. *Computer Security Basics*. O'Reilly & Associates, Sebastopol, California, first edition, 1991.

[200] SAE International. *Guidelines for Development of Civil Aircraft and Systems, Revision A*, 2010.

[201] Johnny Saldaña. *The Coding Manual for Qualitative Researchers.* Sage Publications Ltd, London, 2009.

[202] Tony Sale. Transcript of a lecture given at the IEEE. URL: ⟨`http://www.codesandciphers.org.uk/lectures/ieee.txt`⟩, 18th Febrary 1999.

[203] Christopher Sauer, Li Liu, and Kim Johnston. Where project managers are kings. *Project Management Journal*, 32(4):39–49, December 2001.

[204] Bruce Schneier. *Applied Cryptography.* John Wiley & Sons, Inc., New York, 1996.

[205] Bruce Schneier. *Secrets and Lies: Digital Security in a Networked World.* John Wiley & Sons, New York, 2000.

[206] Science Applications International Corporation Common Criteria Testing Laboratory. Green Hills Software INTEGRITY-178B separation kernel security target, 31 May 2010. Version 4.2.

[207] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.

[208] Claude E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, October 1949.

[209] Jonathan S. Shapiro. Understanding the Windows EAL4 evaluation. *Computer*, 36(2):103–105, February 2003. doi:10.1109/MC.2003.1178059.

[210] Gustavus J. Simmons. Subliminal channels; past and present. *European Transactions on Telecommunications*, 5(4):459–473, July–August 1994.

[211] Gustavus J. Simmons. The history of subliminal channels. *IEEE J. Select. Areas in Comm.*, 16(4):452–462, May 1998.

[212] Michael Spence. Job market signaling. *The Quarterly Journal of Economics*, 87(3):355–374, August 1973.

[213] Patti Spicer. Gaining software assurance through the Common Criteria. *Crosstalk*, 23(5), Sept/Oct 2010.

[214] Rob Spiegel. Plant networks are getting smart about security & safety. *Design News*, 69(7):34–6, July 2014.

[215] William Stallings. *Cryptography and Network Security.* Pearson Prentice Hall, Upper Saddle River, New Jersey, 2006.

[216] Tom Standage. *The Victorian Internet.* Berkley Publishing Group, New York, 1999.

[217] William H. Starbuck. Learning by knowledge-intensive firms. *Journal of Management Studies*, 29(6):713–40, November 1992.

[218] William H. Starbuck. What makes a paper influential and frequently cited? *Journal of Management Studies*, 47(7):1394–1404, November 2010.

[219] Neal Stephenson. *Cryptonomicon.* Avon Books, 1999.

[220] Bruce Sterling. *The Hacker Crackdown.* Bantam Books, New York, 1992.

[221] W. Richard Stevens. *TCP/IP Illustrated, Volume 1: The Protocols.* Addison–Wesley Publishing Company, Boston, Massachusetts, 1994.

[222] George J. Stigler. The economics of information. *Journal of Political Economy*, 69(3):213–225, June 1961.

[223] Joseph E. Stiglitz. The theory of "screening," education, and the distribution of income. *The American Economic Review*, 65(3):283–300, June 1975.

[224] Dorothy Stimson. Dr. Wilkins and the Royal Society. *The Journal of Modern History*, III(4):539–563, December 1931.

[225] Clifford Stoll. *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage.* Doubleday, New York, 1989.

[226] Gary Stoneburner. Toward a unified security/safety model. *Computer*, 39(8):96–97, August 2006.

[227] Gerhard F. Strasser. The rise of cryptology in the European rennaissance. In Karl de Leeuw and Jan Bergstra, editors, *The History of Information Security: A Comprehensive Handbook*, pages 277–325. Elsevier B.V., Amsterdam, 2007.

[228] Sun Microsystems, Inc. *Compartmented Mode Workstation Labeling: Encodings Format DDS-2600-6216-93.* Trusted Solaris 2.5, 2550 Garcia Avenue, Mountain View, California 94043-1100 USA, July 1997. Revision A.

[229] Norman Swindin. *Engineering Without Wheels: A Personal History.* Weidenfeld and Nicolson, 20 New Bond Street, London W1, 1962.

[230] David Talbot. Computer viruses are "rampant" on medical devices in hospitals. *Technology Review*, 17th October 2012.

[231] James J. Tattersall. *Elementary Number Theory in Nine Chapters*. Cambridge University Press, Cambridge CB2 8RU, UK, 1999.

[232] Kathleen Tillotson. Review: *The Man in the Moone and* Nuncius Inanimatus. *The Modern Language Review*, 34(1):92–93, January 1939.

[233] Andrew P. Tobias. *The Funny Money Game*. Michael Joseph Limited, 52 Bedford Square, London WC1, 1972.

[234] Steve Tockey. *Return on Software*. Addison–Wesley Professional, Reading, Massachusetts, 2004.

[235] Johannes Trithemius. Steganographia. *Darmbstadii*, 1499. *Ex officina typographica Balthasaris Aulandri, sumptibus ver Ioannis Berneri, bibliop. Francof.*

[236] Johannes Trithemius. Polygraphiae Librae Sex: Ioannis Trithemii Abbatis Peapolitani, quondam Spanheimensis, ad Maximmilianvm Caesarem. *Coloni: apud Ioannem Birckmannum & Wernerum Richwinum*, 1564. Located in the Christ Church College, Oxford, special collections.

[237] Barbara W. Tuchman. *The First Salute*. Knopf, New York, 1988.

[238] Edward Rolf Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, Connecticut, 1983.

[239] Camille Tuutti. FISMA: Mandating cybersecurity. *Federal Computer Week*, 26(9):32–33, 15th June 2012.

[240] Patrick Tyler. *Running Critical: The Silent War, Rickover, and General Dynamics*. Harper Collins, New York, 1986.

[241] Underwriters Laboratories, Inc. Standard for burglary resistant vault doors and modular panels (UL 608), 2004.

[242] United States Department of Defence. *Trusted Computer System Evaluation Criteria*, December 1985. DOD 5200.28-STD Supercedes CSC-STD-001-83, dtd 15 Aug 83.

[243] United States Department of Defence. *Information Assurance Workforce Improvement Program*. Assistant Secretary of Defence for Networks & Information Integration/Department of Defence Chief Information Officer [ASD(NII)/DoD CIO], January 24, 2012. Incorporating Change 3.

[244] United States Department of Defense. DoD information assurance certification and accreditation process (DIACAP). ASD(NII)/DoD CIO, November 28, 2007. DoD Instruction 8510.01.

[245] U.S. Department of Commerce, National Institute of Standards and Technology. *NIST Special Publication 800-53, Revision 3: Recommended Security Controls for Federal Information Systems and Organizations*, June 2009. Final Public Draft.

[246] U.S. National Security Agency. *Information Assurance Technical Framework version 3.1*, 2002.

[247] Dmitri van Heesch. *Doxygen: Generate documentation from source code*, 2013. URL: ⟨`http://www.stack.nl/~dimitri/doxygen/`⟩.

[248] Mark Vanfleet and Michael R. Dransfield. Design guidance for the MILS architecture to provide MLS damage limitation. Draft No. 1 (NSA Information Assurance Directorate), September 2003.

[249] W. Mark Vanfleet, Jahn A. Luke, R. William Beckwith, Carol Taylor, Ben Calloni, and Gordon Uchenick. MILS: Architecture for high-assurance embedded computing. *Crosstalk: The Journal of Defence Software Engineering*, 18(8):12–16, August 2005.

[250] "Vengie". Re:Cheating is a bad idea (comment on 'Are There Any Smart E-mail Retention Policies?'). Slashdot, July 26, 2008. URL: `ask.slashdot.org/comments.pl?sid=627087&cid=24353451`.

[251] Gilbert S. Vernam. United States patent no. 1,310,719: Secret signaling system, July 22, 1919.

[252] J.M. Verner, S.P. Overmyer, and K.W. McCain. In the 25 years since the mythical man-month what have we learned about project management? *Information and Software Technology*, 41:1021–1026, 1999.

[253] R.D. Wade, G.P. Cawsey, and R.A.K. Veber. A teleprocessing approach using standard equipment. *IBM Systems Journal*, 8(1):28–47, 1969.

[254] Abraham Wald. A method of estimating plane vulnerability based on damage of survivors. Technical Report CRC 432, CRC, July 1980.

[255] Willis H. Ware. Security and privacy in computer systems. In *Proceedings of the Spring Joint Computer Conference*, pages 279–282, Atlantic City, New Jersey, 1967.

[256] Willis H. Ware. Security controls for computer systems: Report of the defence science board task force on computer security. Technical report, Office of the Director of Defence Research and Engineering, Washington, D.C., 11 February 1970.

[257] André Weimerskirch. Do vehicles need data security? In *SAE 2011 World Congress & Exhibition*, Detroit, Michigan, 12–14 April 2011.

[258] Clark Weissman and Timothy E. Levin. Lessons learned from building a high-assurance crypto gateway. *IEEE Security & Privacy*, 9(1):31–39, Jan–Feb 2011.

[259] Steve Welke. Navigating certification & accreditation (C&A)—or is it security authorization? Raytheon Trusted Computer Solutions, 30th March 2011. Copyright 2011 Raytheon Company.

[260] Steve Welke. Update on assessment and authorization processes for cross domain solutions, in a lecture given 23rd October 2014.

[261] Ron Westrum. *Sidewinder: Creative Missile Development at China Lake*. Naval Institute Press, Annapolis, Maryland, 1999.

[262] William H. Whyte. *The Organization Man*. University of Pennsylvania Press, Philadelphia, 1956.

[263] Osmo A. Wiio. *Wiion lait—ja vähän muidenkin*. Weilin+Göös, Espoo, Finland, 1978.

[264] Reinhard Wilhelm and Daniel Grund. Computation takes time, but how much? *Comm. ACM*, 57(2):94–103, February 2014.

[265] John Wilkins. *Mercvry, or the Secret and Swift Meſſenger:* Shewing, *How a Man may with* Privacy *and* Speed *communicate his* Thoughts *to a Friend at any diſtance*. Printed by *I. Norton*, for *Iohn Maynard*, and *Tomothy Wilkins*, and are to be ſold at the *George* in Fleetſtreet, neere Saint *Dunſtans* Church, London, 1641.

[266] John Wilkins. Ibid. *To which is prefix'd the AUTHOR's LIFE, and an Account of his Works. The Mathematical and Philoſophical Works of the Right Reverend John Wilkins, Late Lord Biſhop of Chester.* Printed for J. Nicholſon, at the *King's-Arms* in *Little Britain*; *A. Bell*, at the *Croſs-Keys* in *Cornhill*; *B. Tooke*, at the *Middle-Temple-Gate* in *Fleetſtreet*; and *R Smith* under the *Piazza's* of the *Royal-Exchange*, London, 1708.

[267] John P.L. Woodward. Security requirements for system high and compartmented mode workstations. Technical Report MTR 9992 Revision 1, the MITRE Corporation, November 1987.

[268] Peter Wright. *Spycatcher: The Candid Autobiography of a Senior Intelligence Officer*. Viking Press, New York, 1987.

[269] Jeffrey R. Yost. A history of computer security standards. In Karl de Leeuw and Jan Bergstra, editors, *The History of Information Security: A Comprehensive Handbook*, pages 595–621. Elsevier B.V., Amsterdam, 2007.

[270] Bill Young. Personal communication, 8th September 2014.

[271] Jamie Zawinski. Really bad attitude. URL: ⟨`http://www.jwz.org/gruntle/rbarip.html`⟩, 1998.

Index

Appendices

Appendix A

List of Codes

This is a list of codes used in the grounded theory data analysis of CS-1 and CS-2. The 'Links' column tells how many links a particular code has to other codes in ATLAS.ti; the 'Data' column tells how many quotations in the raw data are associated with each code. Numbers were obtained from the ATLAS.ti version 6.2 'Export selected Codes (XML)' command and edited for formatting. There are two hundred eighteen codes in the list.

Code	**Links**	**Data**
accreditation	295	0
accreditation report	0	1
accreditor	501	4
accreditor model	12	0
AF CDSO	0	1
Akerlof	1	0
analytical memos	50	0
annoyance	2	0
Army	24	0
ARMY CDSO	0	1

Code	**Links**	**Data**
arrogance	17	0
artificial sense of urgency	1	0
belief	0	2
Beta 1	42	0
Beta 2	72	0
bias against	8	1
body language	1	1
building social capital	0	1
C (overseas customer)	0	5
C&A	227	7
CDMO	365	0
CDSO	26	0
CDTAB	198	5
cert_1	77	4
cert_11	2	2
cert_12	2	2
cert_13	8	1
cert_15	3	2
cert_18	1	1
cert_19	1	1
cert_2	3	2
cert_20	21	3
cert_21	1	1
cert_22	1	1
cert_23	18	1
cert_24	1	1

Code	**Links**	**Data**
cert_3	63	3
cert_4	36	4
cert_5	38	1
cert_6	61	3
cert_7	20	1
cert_8	43	1
cert_9	6	1
certification	192	0
certification authority	7	5
certification report	0	1
certifier	53	7
CESG	40	2
Charleston	1	2
cheap	0	1
Common Criteria	0	7
complaint	28	0
compliment	2	1
contractually prohibited	2	0
controlled	0	1
COTR	16	3
cross domain	101	0
CT&E	294	0
D (developer)	728	19
DCID 6/3	0	4
decision	0	1
Decision Making Process	0	2

Code	Links	Data
deniable	0	1
deputy	2	2
dev_1	248	1
W and dev_2	29	1
dev_3	26	1
dev_4	37	1
developer has ultimate experience with	1	0
developer not as good securitywise	1	0
developer thinks Charleston is stupid	1	0
DIACAP	0	4
differing inclinations	16	0
DISA	5	0
distressing	1	2
DNI CAT	38	4
'draconian' approach	1	0
DSAWG	49	3
E (systems integrator)	0	2
email retention	11	0
expensive	0	1
face-to-face	0	2
fact	0	5
FAT	27	0
fiction	0	4
findings	0	1
formal communications	0	5
frustrated	3	0

Code	**Links**	**Data**
frustration	10	0
G (customer's technical adviser)	0	2
H	17	1
harsh words	2	2
high risk	15	0
hopeless	1	0
hotwash	25	0
I	15	1
I173	96	6
I733	41	2
ICD 503	20	0
inconsistency	6	1
informal communications	0	11
interpersonal	3	0
IV&V	148	3
ivv_1	1	1
ivv_2	41	1
J	0	2
K_1	12	2
K_2	0	1
K_3 and dev_7	0	3
knowledge	0	9
L (CC testing laboratory)	0	7
low risk	3	0
M (customer for R-zero)	0	2
malicious	0	1

Code	Links	Data
Marine CSDO	0	1
medium risk	2	0
misunderstanding	1	0
mitigation	94	0
moderates hotwash telecons	0	1
N-prime	333	5
N (CC certifier R-prime)	0	5
Navy	104	0
Navy CDSO	0	2
negative emotional response	0	3
NIAP CCEVS	34	0
NIST SP 800-37	0	2
NIST SP 800-53	0	4
not happy	2	0
official communications	0	1
OGD	0	1
opinion	5	3
P1 (programme office)	0	5
P2 (IV&V contractor)	0	4
pen_1	56	1
pen_2	47	2
pen_3	1	1
pen_4	10	1
pen_5	48	2
pen_6	1	0
penetration tester	0	1

Code	Links	Data
penetration testing	0	1
pentest findings	92	0
pentester	0	8
pentester_related	104	1
perceived risk	1	0
personality	9	0
planned communications	0	3
PMO	0	7
pmo_1	31	2
pmo_2	93	1
pmo_3	23	1
pmo_4	3	1
POA&M	102	1
policies and procedures	52	0
political	11	3
political machinations	0	1
positive emotional response	0	3
praise	4	1
prediction markets	1	0
prejudice	0	2
Process Improvement	0	3
Process Knowledge	0	3
professional security tester secrecy	10	0
Project Knowledge	0	3
project manager	43	1
pushing strongly to stick to schedule	1	0

Code	Links	Data
Q (R-double-prime's certifier)	0	6
R	857	11
R'	0	11
R''	97	10
R-double-prime (product)	0	7
R-double-prime certification meeting	111	0
R-prime (product)	0	6
R-zero (product)	0	4
ratings in perspective	1	0
regression testing	78	3
reports	0	6
role reversal	1	0
Roles	0	7
rumour	18	5
S (system)	0	9
SABI	158	0
schedule	1	1
schedule slip	17	0
sharp remarks	2	1
silent participants	0	5
SP 800-37	7	0
SP 800-53	62	0
ST&E	188	1
STRATCOM	0	3
successful outcome	0	1
tacit communication	4	0

Code	Links	Data
technical risk	48	0
test procedures	0	3
thanked	1	1
timeliness	0	1
tone of voice	0	3
TORA	54	0
training	10	0
TRR	18	0
trying to teach	0	1
TSABI	74	1
turf war	11	0
turnover	14	0
U (talks to S)	0	1
UCDMO	338	0
UCDMO baseline	38	0
uncontrolled	0	1
unfixed finding	1	0
Unknown (from pen_4 report)	0	2
unofficial communications	0	1
unsuccessful outcome	0	1
USN SPAWAR SSC	40	1
V (talks to S)	0	1
we are at a standstill	1	0
X	0	1
Y	0	1
Z	0	1

Ten categories ('families') were defined. The number of unique codes assigned to each and the number of quotations (or analytical memoranda) in each category is shown in the following table:

Category	**Size**	**Data**
accreditors	4	611
certifiers	7	254
CSC	3	50
developers	8	1080
IVV	2	149
NSA173	6	279
OCG	1	21
RM	3	954
RMPMO	7	168
unknowns	7	10

Appendix B

Anonymised Data

The CD-ROM bound into this book contains all of the raw data necessary to replicate the methodology in Chapter 5. The contents of the disc include two directories: `ATLAS.ti` which contains the hermeneutic unit `autocode-CS-1-pkg.hpr6` along with the PDF files comprising the original data; and `WAR`, which contains the researcher's 'Weekly Activity Reports' comprising analytical memoranda. The command line used to generate the ISO 9660 file system on the disk was:

```
$ mkisofs -R -uid 0 -gid 0 -iso-level 4 \
    -V "Supplemental Data" -o supplemental_data.iso \
    supplemental_data/
```

Directory of files on the disc:

```
$ ls -F
ATLAS.ti/      WAR/
$
$ ls -F ATLAS.ti
autocode-CS-1-pkg.hpr6*
backup of autocode-CS-1-pkg.hpr6*
junk/
network_decisions.pdf*
network_developer.pdf*
network_emotional_response.pdf*
```

```
network_grounded_theory.pdf*
network_informal_communications.pdf*
network_official_communication.pdf*
network_r-double-prime.pdf*
network_roles.pdf*
network_r-prime.pdf*
PDFs/
$
$ ls -F WAR
$
0001.txt    0077.txt    0153.txt    0229.txt    0305.txt
0002.txt    0078.txt    0154.txt    0230.txt    0306.txt
0003.txt    0079.txt    0155.txt    0231.txt    0307.txt
0004.txt    0080.txt    0156.txt    0232.txt    0308.txt
0005.txt    0081.txt    0157.txt    0233.txt    0309.txt
0006.txt    0082.txt    0158.txt    0234.txt    0310.txt
0007.txt    0083.txt    0159.txt    0235.txt    0311.txt
0008.txt    0084.txt    0160.txt    0236.txt    0312.txt
0009.txt    0085.txt    0161.txt    0237.txt    0313.txt
0010.txt    0086.txt    0162.txt    0238.txt    0314.txt
0011.txt    0087.txt    0163.txt    0239.txt    0315.txt
0012.txt    0088.txt    0164.txt    0240.txt    0316.txt
0013.txt    0089.txt    0165.txt    0241.txt    0317.txt
0014.txt    0090.txt    0166.txt    0242.txt    0318.txt
0015.txt    0091.txt    0167.txt    0243.txt    0319.txt
0016.txt    0092.txt    0168.txt    0244.txt    0320.txt
0017.txt    0093.txt    0169.txt    0245.txt    0321.txt
0018.txt    0094.txt    0170.txt    0246.txt    0322.txt
0019.txt    0095.txt    0171.txt    0247.txt    0323.txt
0020.txt    0096.txt    0172.txt    0248.txt    0324.txt
0021.txt    0097.txt    0173.txt    0249.txt    0325.txt
0022.txt    0098.txt    0174.txt    0250.txt    0326.txt
0023.txt    0099.txt    0175.txt    0251.txt    0327.txt
0024.txt    0100.txt    0176.txt    0252.txt    0328.txt
0025.txt    0101.txt    0177.txt    0253.txt    0329.txt
0026.txt    0102.txt    0178.txt    0254.txt    0330.txt
0027.txt    0103.txt    0179.txt    0255.txt    0331.txt
0028.txt    0104.txt    0180.txt    0256.txt    0332.txt
0029.txt    0105.txt    0181.txt    0257.txt    0333.txt
0030.txt    0106.txt    0182.txt    0258.txt    0334.txt
0031.txt    0107.txt    0183.txt    0259.txt    0335.txt
0032.txt    0108.txt    0184.txt    0260.txt    0336.txt
0033.txt    0109.txt    0185.txt    0261.txt    0337.txt
0034.txt    0110.txt    0186.txt    0262.txt    0338.txt
```

```
0035.txt    0111.txt    0187.txt    0263.txt    0339.txt
0036.txt    0112.txt    0188.txt    0264.txt    0340.txt
0037.txt    0113.txt    0189.txt    0265.txt    0341.txt
0038.txt    0114.txt    0190.txt    0266.txt    0342.txt
0039.txt    0115.txt    0191.txt    0267.txt    0343.txt
0040.txt    0116.txt    0192.txt    0268.txt    0344.txt
0041.txt    0117.txt    0193.txt    0269.txt    0345.txt
0042.txt    0118.txt    0194.txt    0270.txt    0346.txt
0043.txt    0119.txt    0195.txt    0271.txt    0347.txt
0044.txt    0120.txt    0196.txt    0272.txt    0348.txt
0045.txt    0121.txt    0197.txt    0273.txt    0349.txt
0046.txt    0122.txt    0198.txt    0274.txt    0350.txt
0047.txt    0123.txt    0199.txt    0275.txt    0351.txt
0048.txt    0124.txt    0200.txt    0276.txt    0352.txt
0049.txt    0125.txt    0201.txt    0277.txt    0353.txt
0050.txt    0126.txt    0202.txt    0278.txt    0354.txt
0051.txt    0127.txt    0203.txt    0279.txt    0355.txt
0052.txt    0128.txt    0204.txt    0280.txt    0356.txt
0053.txt    0129.txt    0205.txt    0281.txt    0357.txt
0054.txt    0130.txt    0206.txt    0282.txt    0358.txt
0055.txt    0131.txt    0207.txt    0283.txt    0359.txt
0056.txt    0132.txt    0208.txt    0284.txt    0360.txt
0057.txt    0133.txt    0209.txt    0285.txt    0361.txt
0058.txt    0134.txt    0210.txt    0286.txt    0362.txt
0059.txt    0135.txt    0211.txt    0287.txt    0363.txt
0060.txt    0136.txt    0212.txt    0288.txt    0364.txt
0061.txt    0137.txt    0213.txt    0289.txt    0365.txt
0062.txt    0138.txt    0214.txt    0290.txt    0366.txt
0063.txt    0139.txt    0215.txt    0291.txt    0367.txt
0064.txt    0140.txt    0216.txt    0292.txt    0368.txt
0065.txt    0141.txt    0217.txt    0293.txt    0369.txt
0066.txt    0142.txt    0218.txt    0294.txt    0370.txt
0067.txt    0143.txt    0219.txt    0295.txt    0371.txt
0068.txt    0144.txt    0220.txt    0296.txt    0372.txt
0069.txt    0145.txt    0221.txt    0297.txt    0373.txt
0070.txt    0146.txt    0222.txt    0298.txt    0374.txt
0071.txt    0147.txt    0223.txt    0299.txt    0375.txt
0072.txt    0148.txt    0224.txt    0300.txt    0376.txt
0073.txt    0149.txt    0225.txt    0301.txt    0377.txt
0074.txt    0150.txt    0226.txt    0302.txt    0378.txt
0075.txt    0151.txt    0227.txt    0303.txt    0379.txt
0076.txt    0152.txt    0228.txt    0304.txt    word_counts
$
```

Appendix C

De-anonymisation Codes

To protect the confidentiality of certain participants in the case studies and in view of the classified nature of some of the projects studied, the following code names have been used for projects (Table C.1), organisations (Table C.2), and individuals (Table C.3) throughout Chapters 4 and 5.

Code	**Meaning**	**Role**
R	Radiant Mercury (RM)	cross domain solution
R''	RM version 5.0	a later version of R
R'	Radiant Trust Gold 1.0 (RTG)	a subsystem of S
R_0	AEHF MCS security guard	an earlier version of R'
S	Soothsayer	customer C's project
U and V	Other classified systems outside the boundary of S	*e.g.*, BOWMAN
X, Y, and Z	Other subsystems of S	Windows 2000 servers built by E

Table C.1: Code names for projects.

Code	**Meaning**	**Role**
A	United Kingdom of Great Britain and Northern Ireland	nation in which customer C is located
B	United States of America	developer D's nation
C	U.K. Ministry of Defence	customer

Code	Meaning	Role
		requesting S
D	Lockheed Martin Integrated Systems and Global Solutions (IS&GS)	software developer of R, R_0, R', and R''
E	Lockheed Martin Systems Integration (LMSI)	systems integrator and developer of S
F	United States Air Force	interim government programme office for R
G	U.K. Communications–Electronics Security Group (CESG)	C's technical adviser
I_2	Accenture	IV&V contractor for R''
L	Computer Sciences Corporation (CSC)	Common Criteria validator of R'
M	Advanced Extremely High Frequency (AEHF) Mission Control Segment (MCS)	a previous Common Criteria project in 1999
N	National Information Assurance Partnership (NIAP) Common Criteria (CC) Certification and Evaluation Scheme (CCEVS)	Common Criteria evaluator of R'
N'	U.S. National Security Agency (NSA)	DoD Information Assurance Certification and Accreditation Process (DIACAP) certifier
O	National Reconnaissance Office (NRO)	original government programme office of R and R_0
P_1	United States Navy Space and Naval Warfare Systems Centre	government programme office of R, R', and R''
P_2	Booz Allen Hamilton	IV&V contractor for R'' certification
Q	Defence Information Systems Agency (DISA)	DIACAP co-certifier of R''
competitor_1	Raytheon	one of D's competitors in the CDS market
competitor_2	Boeing	another of D's competitors

Code	Meaning	Role
competitor_3	General Dynamics (GD)	another of D's competitors
competitor_4	ISSE Guard[1]	another of D's competitors
competitor_5	Trusted Manager (TMAN)[2]	another of D's competitors
competitor_6	BAE Systems	another of D's competitors

Table C.2: Code name for organisations.

Code	Meaning	Role
cert_1	Corinne Castanza	NSA I173 (N') and IC CIO/ICIA CAT
cert_2	Calleen Torch	DNI/IC CIO/ICIA/CAT Chief
cert_3	Phyllis Lee	head pentester
cert_4	Emily Martinez	NSA I173 (N')
cert_5	Frank Sinkular	certifier
cert_6	Robert Drake	certifier
cert_7	Dave Oshman	NSA I173
cert_8	Dan Nichols	certifier
cert_9	Paul Livingston	DSAWG Chair
cert_10	Dave Wallick	later head of NSA I173
cert_11	Dave Bowman	U.S. Army CDS Office (CDSO)
cert_12	Rick Perkins	U.S. Air Force CDSO
cert_13	Don Flint	DNI CAT
cert_14	Boyd Fletcher	NSA
cert_15	Glenn Learn	CSTG
cert_16	Don Capanero	certifier
cert_17	Lisa Ackerman	certifier
cert_18	Elizabeth Jubinski	certifier
cert_19	Erika Sollers	certifier
cert_20	Ian Levy	CESG
cert_21	James J. Crandall	certifier
cert_22	John William Ferguson	certifier
cert_23	Patti Spicer	CSC Corp
cert_24	Phillip E. Romans	certifier
cert_25	Maureen Branch	certifier

[1]ISSE Guard is a U.S. Air Force project.
[2]TMAN is a Lockheed Martin product.

Code	Meaning	Role
user_1	Jonathan Scott	STRATCOM Western Region
user_2	Jim Gucken	(unknown)
user_3	Phil [last name unknown]	(unknown)
user_4	Tyler Sipes	NORAD USNORTHCOM HQs
pen_1	Charissa C. Robinson	penetration tester
pen_2	Atri Amin	penetration tester
pen_3	Galina McKee	penetration tester
pen_4	Dana Pipkin	penetration tester
pen_5	Larry Sampson	penetration tester
pen_6	Dave Moran	penetration tester
pen_7	Kevin Gallacchio	penetration tester
dev_1	Kevin R. Miller	software engineer at D
dev_2 and W	Russ Savage	senior R software engineer at D
dev_3	Ian McGlothlin	software engineer at D
dev_4	Larry S. Brown	senior software engineer at D
dev_5	Eric Chiu	staff software engineer at D
dev_6	Kim Frey	deputy project manager at D
dev_7	Steve Bean	senior executive at D
sysadmin_1	Jackie Pockrandt	system administrator at D
H	H. Forsberg	validator's technical representative at L
I	C. Nightingale	validator's project manager at L
ivv_1	Geoff McGarrigle	IV&V contractor
ivv_2	Paul Ozura	Test Director
J	J. Loughry	D's technical representative to G
K_1	S. Steinberger	developer's project manager[3] at D
K_2	C. Grant	D's project manager[3]
K_3 and dev_7	O. Kjono	D's programme manager
dev_6	Craig Christensen	D's Deputy Programme Manager
pmo_1	Dan Griffin	RM PMO COTR at P_1
pmo_2	Dennis Bowden	Programme Office staff

[3]There were several project managers on the R' project; see Chapter 4.

Code	**Meaning**	**Role**
pmo_3	Orville Brown	Programme Office staff
pmo_4	Baasit Saijid	Programme Office staff
T	T. Marso	senior *R* software engineer at *D*
test_1	Kori Phillips	Test Lead at *D*

Table C.3: Code names for individuals.

. . .τῷ Ἀσκληπιῷ ὀφείλομεν ἀλεκτρυόνα· ἀλλὰ ἀπόδοτε καὶ μὴ ἀμελήσητε.