

Motivating Security Engineering with Economics: A Utility Function Approach

Chad Heitzenrater^{*†}

chad.heitenrater@cs.ox.ac.uk

^{*}U.S. Air Force Research Laboratory
Information Directorate
525 Brooks Road
Rome NY 13441, USA

Justin King-Lacroix[†]

justin.king-lacroix@cs.ox.ac.uk

Andrew Simpson[†]

andrew.simpson@cs.ox.ac.uk

[†]Department of Computer Science
University of Oxford
Wolfson Building, Parks Road
Oxford OX1 3QD, UK

Abstract—Establishing the correct mix of functionality and security is key to developing resilient systems; an imbalance will result in system failure, either in system objective or at the hands of an adversary. We present a methodology for reasoning about secure design using economic expressions. We employ Wireless Personal Area Network (WPAN) devices and the IEEE 802.15.4 specification to demonstrate how a utility-based representation can be used to analyse these competing concerns, leading to designs that can be optimised to meet resiliency objectives.

I. INTRODUCTION

Computer security continues to be a vexing problem, as evidenced by the continuing stream of headline-making failures¹. These failures erode our confidence and diminish our privacy, making it harder for the average person to rely on computing devices. Given that ‘perfect’ security (i.e. defence against all possible threats) is known to be cost-prohibitive (if not impossible), developers must carefully weigh security against functional and non-functional requirements to construct systems that balance these often competing demands [1]. Throughout the varied definitions for cyber resilience, a common theme is the need for the system to operate to completion of its purpose — requiring a delicate balance between security and functionality.

We investigate the implications security design decisions made during development have on resiliency and system operation, through the lens of information security economics. Understanding security as a continuum, rather than as an absolute, we seek insight into how decisions throughout the project — especially those during inception and development — affect the overall system success. We argue that a rational, purposeful selection within this continuum can have a profound impact on overall system security and resiliency. To make this case concretely, we examine a class of WPAN devices, along with their associated standards and implementations, as a test case allowing us to identify deeper motivations behind security decisions. The model used to support this argument is widely applicable, potentially informing broader security practice on the Internet at large.

¹A summary of major breaches is given by the *Breach Level Index* (<http://breachlevelindex.com/#!breach-database>).

II. BACKGROUND: IOT SECURITY

The Internet of Things (IoT) has been described generically as “the interconnection of highly heterogeneous networked entities and networks” [2]. To date this has primarily focused on the pervasive instrumentation of physical objects with sensors and actuators, and the connection of those sensors and actuators to the Internet [3]. Examples of applications of such networks include building monitoring/control, utility metering, and surveillance/security, consisting of nodes that may be battery-operated or battery-less connecting to a back-end system for collection and processing of data. Increasingly, this includes systems used in critical infrastructure. The nodes comprising such networks are differentiated from the general Internet by low-power, low cost-electronics, with concomitantly low performance. Their primary means of connectivity are generally based on the IEEE 802.15.4 specification [4] (updated in 2006 [5]), known commonly as Low-Rate Wireless Personal Area Networks (LR-WPANs). Unlike the Internet at large, these resource constraints curtail the use of cryptographic primitives and introduce new challenges to security.

The IEEE 802.15.4 standard specifies a complement of cryptographic suites to meet the varied security needs; these are listed in Table I. The Advanced Encryption Standard (AES) is the most common cryptographic cipher employed to satisfy these needs. Although not all applications requiring security make use of these exact provisions, nearly all use the same set of cryptographic tools. Within these modes, only the NULL and AES-CCM-64 suites are required for compliance with IEEE 802.15.4 [4], [5]. While AES can employ two different key sizes (128- and 256-bit), WPAN hardware implementations exclusively implement the smaller key size for computational and storage reasons. Additionally, while both the CBC-MAC and CCM modes permit the choice of 32, 64 or 128 bits of integrity-protection data, the shorter two modes do not provide a concomitant reduction in computational demands: the full 128 bits are always generated, and then truncated accordingly. The only difference is therefore the number of bytes transmitted.

The security provided by IEEE 802.15.4 is not infallible.

Suite Name	Security Concern	Description
Null	None	No encryption or integrity
AES-CTR	C	Encryption only, counter mode
AES-CBC-MAC-128	I	128 bit CBC-MAC
AES-CBC-MAC-64	I	64 bit CBC-MAC
AES-CBC-MAC-32	I	32 bit CBC-MAC
AES-CCM-128	C and I	128-bit CBC-MAC, AES-CTR
AES-CCM-64	C and I	64-bit CBC-MAC, AES-CTR
AES-CCM-32	C and I	32-bit CBC-MAC AES-CTR

TABLE I
IEEE 802.15.4 SECURITY SUITES (ADOPTED FROM [6]).

In [6], the 2003 version of the security specification was analysed and critiqued, with a number of “defects” identified. Despite these remaining largely unaddressed, IEEE 802.15.4 has seen wide adoption in a range of environments. However, an investigation of the trade-offs inherent in selecting the operational parameters of an IEEE 802.15.4 network remains absent from the literature. A goal of this paper is to explore the potential ramifications of the cryptographic modes to resiliency, employing an economic argument.

III. MODELLING FOR DEVELOPMENT

While much has been written regarding the analysis and optimisation of security from an enterprise (post-deployment) perspective, there exists little work examining functionality, security and resiliency during system development. To facilitate an examination of these issues within the context of WPAN devices, a lightweight model inspired by [7] is employed, comprising the following steps: *Security Requirements and Analysis*, *Security Design and Implementation*, and *Operations and Maintenance*. This is extended with utility-based requirement definitions that permit analysis beyond that of [7] and consider security, resiliency and functionality as trade-offs throughout the system life-cycle.

A. Security Requirements and Analysis

Any secure development life-cycle starts with a definition of the threat model and relevant security policy(ies). Often this culminates in a risk analysis, despite the difficulties of reliably estimating and correctly quantifying such values [7]. This work replaces this final step with the development of a parameterised, utility-based definition of a security context. In addition to bypassing the perils of risk estimation that underlie many information security investigations, this will inform later analysis and provide insight into the post-development considerations to come.

1) *Security Requirements and Policy*: Despite growing recognition that it is imperative to account for security when software systems are designed and developed [8], security requirements processes have yet to find a common form and structure, and are particularly disjoint in the extent to which concrete measures are stated [9]. Of particular concern is the practice of specifying a solution in the requirements phase [10], often in the form of a specific technology [11] as this undermines the engineering process.

An obvious requirement is that the system comply with the IEEE 802.15.4 specification. While it is the security implications of this requirement that are of interest to this analysis, this decision involves more than merely security considerations (e.g. interoperability, marketing, or compliance). While we assume this decision, the analysis applies to other constraints equally well. This decision imparts a choice of modes, as specified in Section II (to include ‘none’). Such decisions impart considerations often left to the engineering process: the specific cryptographic method, bit length, and implementation medium (hardware or software), for example. Looking to requirements for guidance is likely to yield insight, as typical security requirements take a vague form. Common applicable examples include *integrity* (“The application shall prevent the unauthorized corruption of all communications passing through networks that are external to any protected data centers” [11]) and *confidentiality* (“The application shall not allow unauthorized individuals or programs access to any communications” [11]).

While such requirements communicate common security principles, they lack the specificity required to quantifiably assess whether they have been fulfilled. They serve as examples of security defined as a set of higher level goals devoid of the specific guidance required to render the requirement in a form that is sufficiently specific and verifiable to guide designers [12]. Processes such as the Atlantic Systems Guild Volere Process² seek to address this problem through the use of ‘fit criteria’, augmenting functional requirements that are otherwise too vague or ambiguous for direct implementation [13]. Our approach will use utility-based constraints on the system in a similar manner in order to provide the necessary specificity.

The choice of IEEE 802.15.4 mode inherently addresses confidentiality (AES-CTR and AES-CCM), integrity (AES-CBC-MAC and AES-CCM), neither (NULL), or both. (See Table I.) Weighted against policy and functional requirements, dimensions of availability and resiliency are brought to light.

Confidentiality. Bit length of the block and the key are a typical focus of security analysis relative to confidentiality. However, this is removed from consideration by way of the standard’s fixed parameters. Additionally, the value of encryption without authentication is questionable, with some suggesting the removal of AES-CTR from the standard [6].

Integrity. Considering integrity, the number of bits used in message authentication is of interest with alternative modes providing a trade-off with functionality. Beyond the obvious transmission savings, little guidance is provided on the conditions under which these modes should be chosen.

The requirement of IEEE 802.15.4 compliance drives security and resiliency decisions. An insecure approach of maximising availability through the use of the NULL mode has a negative effect on resiliency as the system is susceptible not only to attack, but also to any verification failure resulting from attack or transmission error. Choice of any other mode

²See <http://www.volere.co.uk/>.

will subsequently result in varying impacts to security and resiliency, both positive and negative, as the system's functionality and lifespan are impacted. This requires considerations beyond confidentiality, integrity and availability, to include functional concerns: throughput, time between maintenance, and product lifetime. This task of correctly defining hardware, software and protocol designs that implement the stated functional and security requirements often falls to the system designer, who may (or may not) understand such constraints. Framing these functional and non-functional considerations in light of actions necessary for security and resiliency, we derive key considerations.

First, current practice suggests any and all security measures should be employed. Encryption and authentication of all communications is often a matter of policy regardless of the nature of the data, with incorporation of encryption presented as a matter of course in [7]. For WPAN devices, the motivation for such measures is less clear. While more secure, the negative impacts on system function call such assumptions into question. We ask, for a given set of attacker assumptions, *can choosing a 'lesser' cryptographic mode make WPAN devices more secure?*

Second, while the AES key space is fixed, the MAC employed is a design choice. The IEEE 802.15.4 specification permits three options: 128-, 64-, and 32-bit MACs for CBC-MAC and CCM modes. The reality is nuanced, as in each case the entire 128-bit MAC is generated. As a result this choice does not result in any reduction of processing resources. Shorter MAC addresses concern only network transmission costs, at the increased risk of an attack undermining data integrity. We examine *if and how the conditions manifest that result in a rational choice of a 64- or 32-bit MAC mode.*

Third, implementation of the system in hardware creates a series of trade-offs. Implementation of the 64-bit CCM mode and NULL cipher suites is sufficient to meet IEEE 802.15.4 compliance, and can be accomplished in hardware, software, or a combination of the two. We explore this decision, with the aim of identifying *the conditions that lead to the rational choice of designs that meet the specification but for which security is weighed against other constraints.*

To analyse these questions requires the establishment of a *computable security context*, based in utility theory, to aid decision-making.

2) *Utility Definition:* Often, security decisions are driven by high level risk-based analysis, as advocated in [7]. Applying such an approach to functional design can be perilous, as the connection between such assessments and technical decisions are often unclear. Instead, a utility-defined model is derived from the requirements of Section III-A1 to examine these questions.

From these requirements the managerial, operational and functional aspects of the system are derived in order to examine the security, resiliency and functionality trade-space. Focusing on WPAN devices employed for monitoring and control, a highly simplified model can be constructed. Such a

'device' (D) is defined as set of hardware (HW) and software (SW):

$$D = HW + SW \quad (1)$$

Depending on the specific application, additional functional requirements come into consideration, each impacting security and resiliency. Examples include: the number of years the device is intended to operate (Y) — with a finer grained specification considering time (t) as a subdivision of Y ; the frequency with which the device transmits (f); and the size of the data payload to be transmitted (m).

Such requirements are often the result of a requirements analysis, independent of security considerations, set by necessity or by customer edict (e.g. 'the message payload is to be 32 bytes'). In many instances, such as research and development or early product development, specific values are less important than ranges and limits. Where possible, the parameterisation of these values will permit a broader consideration of potential solutions.

A number of non-functional requirements and constraints require consideration. For instance, device location impacts 'power' (P) available to the device; it may also inform the cost to replace or maintain the device. When examining the impact of design decisions on the system life-cycle, planning for such considerations is necessary to enable resiliency.

a) **Managerial Requirements:** Starting at the most conceptual level, we first examine the managerial aspects in order to derive parameters to use in our utility definitions. As with any development effort, the primary management concern surrounds the minimisation of cost (C), which must be kept below a target budget α .

$$C = c_p + c_d + c_m \quad \text{and} \quad \min(C) < \alpha \quad (2)$$

Here, c_p refers to the cost of power, c_d refers to the per device cost,

$$c_d = c_{HW} + c_{SW} \quad (3)$$

and c_m refers to the device maintenance cost over the system's lifetime of Y years. Therefore, minimisation of C directly relates to the minimisation of c_p , c_d , and/or c_m relative to Y .

b) **Functional Requirements:** The identified security requirements can be characterised by the resources provided to them. In this case, the resources related to WPAN devices are relatively straightforward, constrained only by a handful of functional aspects already defined by the environment.

Output (O) includes network traffic with packets of size x produced by a device at a rate r , in turn expressed in terms of a period of time t and frequency f .

$$\begin{aligned} O(x, r) &= x \cdot r & \text{where} \\ x &= m + m_o + m_s & \text{and} \\ r &= \frac{f}{t} \end{aligned} \quad (4)$$

The values m_o and m_s relate to the message overhead incurred by the choice of protocol (overhead) and security, respectively, for a given message size m . These relationships are naturally

bounded by the constraints of any chosen hardware or software, such that:

$$\begin{aligned} O(x, r) &< O_{max} \quad \text{and} \\ x &< x_{max} \end{aligned} \quad (5)$$

Here, O_{max} is the capacity of the network in time frame t . The packet capacity x_{max} and the network O_{max} is established by the choice of hardware, software and protocol.

Power consumption (P) is considered relative to the energy afforded to the system. Judicious use of resources directly affects the maintainability, user experience and the direct costs to manufacture, produce and sell the product — not to mention the ability of the device to sustain attack and execute its functional purpose.

Generally, the minimisation of power consumption to maximise operating capacity and lifetime is a design goal for constrained systems. This can be accomplished either by the power allocation being sufficient to meet the lifetime expectancy of the product, or with a sufficiently low service cost to maintain the necessary power source. We define the total power required $P_{required}$ to be the combination of the power required for security ($P_s(x)$) and transmission ($P_t(x)$) operations, for a given packet size x . This must then be accounted for given the output of the device O over its lifetime, Y .

$$\begin{aligned} \min(P_{required}) \\ P_{required} = (P_s(x) + P_t(x)) \cdot r \cdot Y \end{aligned} \quad \text{and} \quad (6)$$

It should be obvious, then, that the required power is ideally lower than the total power available to the device (P_{total}):

$$P_{required} < P_{total} \quad (7)$$

Failure to meet this latter constraint will thus incur a necessary maintenance cost, as described below.

A definition for security utility can be derived from the dual confidentiality and integrity requirements, expressed as the relationship to the adversary capability and derived from key phrases in the requirements: “shall prevent” and “shall not allow” are interpreted as “not possible within the lifetime of the device”. If we define a function $E()$ to represent the time required for an adversary to undermine a particular security attribute (represented by $-$), we can establish the following utility definitions for a security overhead, m_s .

$$\begin{aligned} E_{-conf.}(m_s) &> Y \quad \text{and} \\ E_{-int.}(m_s) &> Y \end{aligned} \quad (8)$$

The choice to define this utility in terms of the system lifetime is both a result of the terminology used in the specification of the security requirement, as well as an interpretation that defines criteria to bound the vagueness of the requirement. Alternative interpretations capturing different aspects of resiliency are possible; for instance, effort to undermine confidentiality could be greater than the lifetime of the data, or could be greater than a sufficiently large computational expenditure (as is commonly employed in traditional cryptographic security). This changes the utility function and its relationship to the

functional requirements, potentially leading to a different construct that may incur other trade-offs in terms of functionality, flexibility, or cost.

This approach is a departure from previous attempts to define security strictly in economic terms. Efforts such as [14] require the explicit estimation of adversary considerations, such as “successfulness [sic] of attack” and “motivation to attack” to form a Return on Information Security Investment (ROISI) [14]. Other approaches utilise the more traditional Return on Security Investment (ROSI) [15] and Return on Attack (ROA) [16], which are rooted exclusively in monetary terms and restrict the view to purely financial terms — which is problematic for intangible assets. These approaches require fine-grained, risk-based estimation of quantities, which (put mildly) is “no simple task” [15].

c) Operational Requirements: For this analysis we limit our consideration of the operational utility to the costs of maintenance relative to the sustainment of the functional requirements identified above. When power requirements for the device exceed the total power available ($P_{required} > P_{total}$) a maintenance cost will be incurred in order to replace the power supply at the point that the power has been depleted. For a system that is attached to the electrical grid this is not a consideration; however, for a detached scenario the cost of maintenance is defined as:

$$c_m = \begin{cases} 0 & \text{where } P_{total} > P_{required} \\ \frac{P_{total}}{P_{required}} \cdot \delta & \text{otherwise} \end{cases} \quad (9)$$

Here, δ represents the estimated cost of a single maintenance event. This is specific to a given system, supporting the incorporation of business costs in the analysis.

These utility functions, along with the requirements they encapsulate, define the context against which we measure the security and resiliency of the implementation.

B. Security Design and Implementation

Secure development practices involving the use of patterns, modelling, review, language choice, and bug analysis are essential to good security practice, as the address of vulnerabilities is an imperative shared by all software vendors [17]. We take these as a given in any engineering environment, and instead focus on design in the face of conflicting, or even incompatible, requirements.

1) Hardware versus Software: Starting with the examination of hardware versus software implementation for compliance with IEEE 802.15.4, a series of considerations arises. Purpose-made hardware is readily available, but incurs an expense and inflexibility (e.g. the exclusive use of 128-bit AES) that may render software an attractive option, especially in environments where the system functionality or security construct requires specialisation, or is likely to change during its lifetime. It has been recognised that price, resources and efficiency are factors that must be considered at the design stage [18]. Employing the utility-based definitions of Section III-A2, we can derive simple relationships that result in insight into such design considerations.

Operation	Hardware Accel.		w/o Hardware Accel.	
	Time	Energy	Time	Energy
AES-128 encryption (single block)	0.09 ms	2.43 μ J	4.89 ms	131.89 μ J
AES-CCM encryption/decryption (96 bytes)	0.81 ms	21.85 μ J	37.24 ms	1.01 mJ
AES-CCM MAC generation (96 bytes data + 11 bytes AD)	0.61 ms	16.45 μ J	54.77 ms	1.48 mJ
802.15.4 frame transmission (115-byte payload + 11-byte header)	4.93 ms	424.47 μ J		

TABLE II
MEASUREMENTS OF CRYPTOGRAPHIC OPERATIONS ON A CC2530 SYSTEM-ON-A-CHIP (96 BYTE PAYLOAD).

Starting at the functional level with the consideration of power, the defined constraints in Equation 6 can be combined to derive a single power utility relationship:

$$\min(P_{required}) = \min((P_s(x) + P_t(x)) \cdot r \cdot Y) \quad (10)$$

We consider this relationship in light of the hardware-software trade-off, examining the effect of P_s (the power devoted to security) on $P_{required}$ (the overall power allowance). Table II shows measurements acquired using a CC2530³ system-on-a-chip.

From this data, an estimate can be made of the impact of the wider range of IEEE 802.15.4 cryptographic modes on power consumption. Using a message length of $m = 96B$ (to accentuate the differences), we perform an examination of hardware and software using 1 message per second, over 1 year ($r = \frac{1}{1s}$ and $Y = 31536000s$). Recall that the hash generated for AES-CBC-MAC and AES-CCM is always 128 bits; the difference in modes (32-bit, 64-bit, and 128-bit) relates to the amount of the MAC transmitted. At this message size and data rate the merits of employing the hardware encryption is blatantly visible, when comparing the same algorithm on larger packet sizes. In AES-CCM mode, the energy consumption of software is shown to now be *65 times* that of hardware.

Considering now the managerial requirements, we again combine the individual defined utilities to arrive at a single relationship to drive the design choice:

$$\min(C) = \min(c_p + (c_{HW} + c_{SW}) + c_m) \quad (11)$$

From this relationship we see the impact of this data on each term. The obvious — and most often the primary — driver relates to the up-front costs of the device (all else being equal in terms of security functionality). Here, we find little difference; employing an open-source library may seem an attractive option, especially if we assume the hardware-software interface to be of roughly the same magnitude as the incorporation of an open source library. The 8-bit hardware devices retail for roughly £100, with 128 Kb of flash memory and a maximum operating frequency of 16 MHz. Newer hardware with additional memory and a higher operating frequency will demonstrate better performance, albeit at an increased per-unit cost. We find more differentiation relative to c_p , where the software solution results in 16.3 times the energy needs in the AES-CTR operation, and *65 times* for the

full AES-CCM stack. However, the effect of this disparity on c_p will be relative to the employment scenario and subsequent available power source; a device connected to the power grid will result in value inherently different than a battery-driven sensor. It is in considering maintenance (c_m) that we see a strong differentiation, tightly related to employment scenario. From Equation 9, this cost is driven by the maintenance frequency (expressed in terms of the power required and power available), as well as the magnitude per event, δ . While δ is strictly driven by use cases, the frequency resulting from this choice is easily derived. In our test-bed, the CC2530 boards operated on two AA batteries providing roughly *15kJ*. In this configuration, the software implementation results in a replacement rate of more than *five times per year* in the case of AES-CCM.

2) *Cryptographic Mode*: Examination of the various modes yields the relationship between the mode and the incurred cryptographic overhead m_s , for a payload size m bytes and m_o byte header, $p = m + m_o + m_s$ (Equation 4):

$$\begin{aligned} \text{NULL, AES-CTR} &= m + m_o \\ \text{AES-CBC-MAC, AES-CCM} &= m + m_o + \{4, 8, 16\} \end{aligned}$$

For the NULL suite it should be clear that no effort on the part of the adversary is required. Similarly, and as identified in [6], the AES-CTR mode fails to offer integrity protection and therefore is susceptible to such attacks. Neither contribute to the security or resiliency of the system, and further consideration is limited to the AES-CBC-MAC and AES-CCM modes.

a) *Choosing MAC size*: All IEEE 802.15.4 radios have a theoretical maximum raw data rate of 250kb/s on network by specification [4]. Combining the minimum packet size for each mode resulting from a 1-byte message, a theoretical upper bound for an adversary to brute-force the submission of a valid packet can be derived — ignoring the obvious service denial that would result. The top row of Table III provides a lower bound estimate of the minimum average time for an adversary to theoretically spoof a single packet in such a configuration.

A more likely scenario is the submission of a packet to a collection node receiving data from such devices. It is reasonable to expect such a system to have a significantly higher data rate (potentially accepting data from numerous devices). Similar calculations can be performed for an IEEE

³<http://www.ti.com/product/CC2530>

Data Rate	Ave time to success (Years)		
	32 (2^{31})	64 (2^{63})	128 (2^{127})
250kb/s	0.0349	$1.872e + 8$	$4.834e + 27$
7Gb/s	$1.245e - 6$	6685.056	$1.726e + 23$

TABLE III
THEORETICAL MINIMUM AVERAGE TIME (IN YEARS) TO BRUTE-FORCE
THE SUBMISSION OF A VALID 1-BYTE PACKET.

802.11ad device, which has a maximum theoretical speed of 7 Gb/s⁴ (see the bottom row of Table III).

Clearly, the 64-bit version (the only AES-CCM mode required by the specification) is sufficient to withstand such a hypothetical attack. However, the 32-bit mode, at first appearing inadequate, may deserve further analysis. The range of values above correspond to more than 305.4 hours (or 12 days) to less than a minute (0.654 minutes) at 802.15.4 and 802.11ad data rates, respectively, at the theoretical limit.

Adopting an ‘attacker economist’ assumption⁵ raises the question as to the benefit derived from a single spoofed packet in terms of the value of hours, or even days, of time necessary. It is at this point that concrete scenario- and adversary-specific approaches such as attack trees [19] or misuse cases [20] can lead to measurements such as Return on Attack (ROA) [16]. The benefit of the utility-driven approach is in the ability to perform analysis generally, without relying on such estimates until a meaningful, well-defined scenario is constructed.

b) Non-CCM modes: A common rationalisation is ‘why risk it?’ Given that the MAC will always be computed as 128 bits and then truncated, what (if any) gain results from employing the 32- or 64-bit form — especially given the potential downside, which includes reduced hardware support for the non-required modes? For the purposes of this application, latency serves as a proxy for power consumption (and may be constrained by a functional requirement in its own right); here it permits the variance between the different nodes to be more clearly investigated.

Figure 1 shows the variance in the latency for a single-hop, round-trip packet transmission between two CC2530 nodes within our test bed over a series of message sizes (m). This data was generated by measurement of the overall latency (averaged over multiple runs), and subtracting from that the time incurred by the initialisation vector generation, the software stack, and the transmission (also averaged over multiple runs) — thereby representing only the latency incurred by the cryptographic process. The header size (m_o) is held constant at 11 bytes⁶.

What is immediately striking in this graph is the consistency in the modes. As the packet size increases, the AES-CBC-MAC (integrity) modes show slow growth consistent with the packet size, as does AES-CCM, although at a higher rate of increased latency. This is reflective of the additional overhead

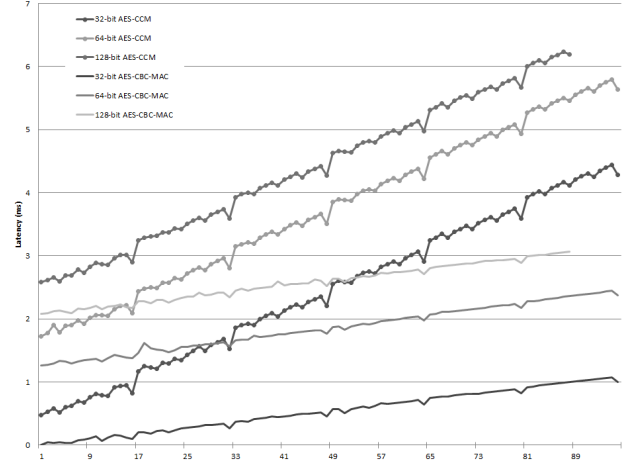


Fig. 1. Latencies incurred by cryptographic operations at different packet sizes. This graph shows round-trip latency, with the cryptographic functions performed four times between two nodes, A and B : encrypt request at A , decrypt request at B , encrypt reply at B , and decrypt reply at A .

Operation	HW Accel. Energy (kJ)	w/o HW Accel. Energy (kJ)
AES-CTR	0.45979	24.95570
AES-CBC-MAC	0.51877	46.67328
AES-CCM	1.20783	78.52464

TABLE IV
POWER (ENERGY) CONSUMPTION FOR HARDWARE ACCELERATED AND
NON-HARDWARE ACCELERATED IMPLEMENTATIONS OF IEEE 802.15.4
CRYPTOGRAPHIC MODES.

AES block encryption incurs (visible in the data in Table IV). Given this encryption overhead, it should be unsurprising to find that the AES-CCM will by and large place a larger demand on system resources. However, when analysed against the system size there exist points where CCM modes are more efficient than their counterparts — likely attributable to shared cryptographic operations that are accelerated on certain devices. Given this insight, design spaces can be identified where specific cryptographic modes represent a *more* efficient choice:

- 1) 64-bit AES-CCM, compared to 128-bit AES-CBC-MAC (for messages under 16 bytes).
- 2) 32-bit AES-CCM, compared to 64-bit AES-CBC-MAC (for messages under 32 bytes) and 128-bit AES-CBC-MAC (for messages under 48 bytes).

C. Operations and Maintenance

Investigations into the implications of security choices in the development process often stop short of providing insights into operations and maintenance, citing a preference to “focus on the steps directly related to development” [7]. Employment of utility-based definitions permits the consideration of overall system security, leading to an analysis of capability and providing insights into the design and implementation that were previously non-obvious and overlooked. The result of this

⁴According to <http://standards.ieee.org/news/2013/802.11ad.html>.

⁵As defined by Daniel Bernstein in <http://ecrypt-eu.blogspot.de/2015/11/break-dozen-secret-keys-get-million.html>.

⁶The 128-bit measurements are not measured beyond 88 bytes, as the maximum size for the packet x_{max} is reached.

analysis leads to tangible design insights regarding hardware versus software implementations, the choice of MAC size, and suitable cryptographic modes.

1) *Choosing a hardware or software implementation:* From Section III-B1, the demonstrated power overhead for software implementation has implications beyond design. Considering the issue of operational maintenance and replacement, this results in the following relationship:

$$\text{replacement rate} \approx P_{\text{required}} \cdot r \div P_{\text{total}} \cdot c_m \quad (12)$$

Employing this relationship permits examination of issues such as increased size of the power source (at a higher cost), employment of different hardware (with better power efficiency), consideration of the data packet size and transmission frequency (to alter O), or the development of hardware cryptography (with potential increased cost of development, or a reduction in flexibility that reduces the lifetime of the device Y). The impacts of design choices beyond the development cycle is not realised in traditional software architecture decision methods (e.g. [21]), which rely on one-time cost, risk estimation and relative rankings to derive a 1-to-n list for investment. Consideration of the economic impact of functional requirements empowers the system and software designer to better balance the security needs with other goals, such as resiliency.

2) *Choosing a MAC size less than 128bits:* In Section III-B2 a simple calculation demonstrated how data rates act as an inhibitor to brute force attacks against message integrity in such systems. While the 32-bit AES-CBC-MAC protocol could theoretically be subverted in a matter of seconds to days, one could conceive of situations where even this level of protection would be found acceptable. A single malformed 1-byte packet is likely insufficient to meaningfully undermine many applications, and could easily be mitigated by time averaging, outlier removal, or other processing on the receiver end. The benefit would be increased operational life, balancing operation of a singular device against resiliency of the overall system.

These decisions depend on the goal and nature of the processing, as well as the application scenario and functional requirements such as the radio bandwidth, message packet size and power requirements. When related to the broader perspective provided by threat modelling and risk analysis, security decisions have the potential to significantly impact other desirable traits, such as system lifetime and performance. Such utility refinements define a security context against which the performance of a given architectural design can be measured. A complementary proposal by Schechter [22] examines the metric of ‘cost-to-break’ (CTB) as a standard measure for security, while Camp and Wolfram [23] propose computing resources as the units by which CTB is measured.

Such analyses highlight an often overlooked aspect of security: “while some investment in security is good, more security is not always worth the cost” [24]. There is no lack of security failures that result from bypassing or undermining security mechanisms, often due to their interference with the

functionality or usability of the systems they were put in place to protect. Considering security in terms of resources within the engineering process permits an approach where they can be balanced against functional requirements, leading to increased system resiliency.

3) *Choosing modes other than AES-CCM:* Section III-B2 further examined the relationship between resource consumption and message size. This examination highlighted how careful design decisions inform both security and overall system operation, finding points where (on the tested hardware) particular suites are most effective. The result is a set of identified circumstances where the default hardware mode (64-bit AES-CCM) might not be optimal:

- *The security provided by AES encryption at a block and key size of 128-bits is inadequate.* While the security provided by this construct is seen by most to be adequate for at least another 10 to 15 years⁷, one can conceive of scenarios where device outputs could have a confidentiality requirement or lifetime Y that exceeds this date.
- *For integrity alone, AES-CCM may be overkill.* Without a confidentiality requirement (e.g. security requirement (1) above), resource savings can be found in most cases by choosing a CBC-MAC mode over CCM (with the exception of 128-bit AES-CBC-MAC for short messages). Conversely, in contexts where confidentiality provides value — or at least does not impede the system operation — the addition of the 32-bit MAC shows better performance for messages under 48 bytes.

This decision becomes more nuanced when considering the wider suite of modes, or when confidentiality is of greater concern. In such cases, the trade-space must weigh functional and security requirements within the context of costs over the device’s lifetime. The defined utilities demonstrate how such tools can be employed in a holistic process, defining the conditions of resiliency.

IV. CONCLUSION

Through the application of software engineering, economics and wireless security principles we have examined the challenge of designing secure, functional systems that balance diverse goals and exhibit resiliency. This investigation utilised the IEEE 802.15.4 specification as a test case, providing a basis to form concrete examples around the WPAN use case. An important aspect to this contribution is its focus on analysis at the design level; where such analysis is not uncommon post-deployment, this approach aims to enable insight before designs are set, and the cost of change is a barrier.

On a direct level, we have shown that various WPAN cryptographic modes defined in IEEE 802.15.4 exhibit specific trade-offs relative to security and functionality. Unlike other contributions (e.g. [6]), we characterised weaknesses in certain modes (e.g. AES-CTR) as representing a specific — if not uncommon — portion of the security trade-space. The existence

⁷Based on the calculations for 128-bit symmetric key lengths at <http://www.keylength.com/en/compare/>.

of such modes necessitates guidance to the community of practitioners, designers and implementers — not all of whom will have a strong understanding of the nature of security. To this end, we have attempted to provide tools by which this trade-space between functionality and security can be quantified and rationalised. This has resulted in specific recommendations concerning hardware versus software design, the choice of MAC size, and when non-AES-CCM modes may be a more appropriate implementation choice.

At a broader level, this line of research has the potential to inform the wider field of information security design. Development of utility-based definitions to accompany security requirements provides a link to traditional requirements elicitation. These relationships between security and the functional, economic and managerial requirements provide practitioners with the means to employ a rational approach — one in which the aim is to bound security investments through quantifiable relationships (e.g. risk management). This permits the consideration of resiliency as a holistic principle combining considerations of the system environment [12], the balance of protection, detection and recovery/response [25], and the relationship with alternative forms of security provision (e.g. deterrence [26]). A *computable security context* provides the basis by which practitioners can understand the security contribution of their designs earlier, and can better inform established SSE practices.

Future work will broaden and formalise the notion of utility-driven security engineering, and understand its relationship with secure software engineering practice. It is the authors' belief that then moving from the notion of absolute security, the definition of a security context becomes essential to understanding the state of security in the system. Methods by which such a context is developed and subsequently informs the entire system life-cycle are necessary in order to align current computer security practice with theoretical, but insightful, findings from information security economics. Applying this concept to a complex system with multiple operational requirements necessitates extension to scalable, multi-dimensional analysis. Naturally, this research is also not the final word in WPAN security; much more can be accomplished to extend this analysis to other contexts (standards, hardware, and functional requirements) in order to understand how our findings extend to the broader class of IoT devices.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their suggestions.

REFERENCES

- [1] P. T. Devanbu and S. Stubblebine, "Software engineering for security: A roadmap," in *Proceedings of the Conference on The Future of Software Engineering (ICSE 2000)*. IEEE / ACM, 2000, pp. 227–239.
- [2] T. Heer, O. Garcia-Morchon, R. Hummen, S. Keoh, S. S. Kumar, and K. Wehrle, "Security challenges in the IP-based Internet of Things," *Wireless Personal Communications*, vol. 61, no. 3, pp. 527–542, December 2011.
- [3] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, October 2010.
- [4] IEEE, "IEEE standard for local and metropolitan area networks — part 15.4: Low-rate wireless personal area networks (lr-wpans)," IEEE Std 802.15.4-2011 ((Revision of IEEE Std 802.15.4-2006), September 2011.
- [5] —, "IEEE standard for local and metropolitan area networks — part 15.4: Low-rate wireless personal area networks (LR-WPANs) amendment 1: MAC sublayer," April 2012.
- [6] N. Sastry and D. Wagner, "Security considerations for IEEE 802.15.4 networks," in *Proceedings of the 3rd ACM Workshop on Wireless Security (WiSe 2004)*. ACM, 2004, pp. 32–42.
- [7] A. Apvrille and M. Pourzandi, "Secure software development by example," *IEEE Security and Privacy*, vol. 3, no. 4, pp. 10–17, July 2005.
- [8] K. Goseva-Popstojanova and A. Perhinschi, "On the capability of static code analysis to detect security vulnerabilities," *Information and Software Technology*, vol. 68, no. C, pp. 18–33, December 2015.
- [9] I. A. Tøndel, M. G. Jaatun, and P. H. Meland, "Security requirements for the rest of us: A survey," *IEEE Software*, vol. 25, no. 1, pp. 20–27, January 2008.
- [10] P. Salini and S. Kanmani, "Survey and analysis on security requirements engineering," *Computers & Electrical Engineering*, vol. 38, no. 6, pp. 1785–1797, 2012.
- [11] D. G. Firesmith, "Engineering security requirements," *Journal of Object Technology*, vol. 2, pp. 53–68, 2003.
- [12] C. Haley, R. Laney, J. Moffett, and B. Nuseibeh, "Security requirements engineering: A framework for representation and analysis," *IEEE Transactions on Software Engineering*, vol. 34, no. 1, pp. 133–153, January 2008.
- [13] S. Robertson and J. Robertson, *Mastering the Requirements Process*, 1st ed. Addison-Wesley Professional, 1999.
- [14] A. Mizzi, "Return on information security investment — the viability of an anti-spam solution in a wireless environment," *International Journal of Network Security*, vol. 10, no. 1, pp. 18–24, January 2010.
- [15] W. Sonnenreich, J. Albanese, and B. Stout, "Return On Security Investment (ROSI) — A practical quantitative model," in *Journal of Research and Practice in Information Technology*, vol. 38, no. 1, 2005, pp. 239–252.
- [16] M. Cremonini and P. Martini, "Evaluating information security investments from attackers perspective: the return-on-attack (ROA)," in *4th Annual Workshop on the Economics of Information Security (WEIS 2005)*, June 2005.
- [17] S. Lipner, "The trustworthy computing security development lifecycle," in *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC 2004)*. IEEE Computer Society, 2004, pp. 2–13.
- [18] M. Botta, M. Simek, and N. Mitton, "Comparison of hardware and software based encryption for secure communication in wireless sensor networks," in *Proceedings of the 36th International Conference on Telecommunications and Signal Processing (TSP 2013)*, July 2013, pp. 6–10.
- [19] B. Schneier, *Secrets & Lies: Digital Security in a Networked World*, 1st ed. John Wiley & Sons, Inc., 2000.
- [20] G. Sindre and A. L. Opdahl, "Eliciting security requirements with misuse cases," *Requirements Engineering*, vol. 10, no. 1, pp. 34–44, 2005.
- [21] R. Kazman, J. Asundi, and M. Klein, "Making architecture design decisions: An economic approach," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU/SEI-2002-TR-035, 2002.
- [22] S. Schechter, "Quantitatively differentiating system security," in *Proceedings of the 1st Workshop on the Economics of Information Security (WEIS 2002)*, 2002.
- [23] L. J. Camp and C. Wolfram, "Pricing security: A market in vulnerabilities," in *Economics of Information Security*, ser. Advances in Information Security, L. J. Camp and S. Lewis, Eds. Springer, 2004, vol. 12, pp. 17–34.
- [24] L. A. Gordon and M. P. Loeb, "The economics of information security investment," *ACM Transactions on Information and Systems Security*, vol. 5, no. 4, pp. 438–457, 2002.
- [25] S. A. Butler, "Security attribute evaluation method: a cost-benefit approach," in *Proceedings of the 24th International Conference on Software Engineering (ICSE 2002)*. IEEE / ACM, 2002, pp. 232–240.
- [26] C. Heitzenrater, G. Taylor, and A. C. Simpson, "When the winning move is not to play: Games of deterrence in cyber security," in *Proceedings of the 6th Conference on Decision and Game Theory for Security (GameSec 2015)*, ser. Lecture Notes in Computer Science, vol. 9406. Springer, 2015, pp. 250–269.