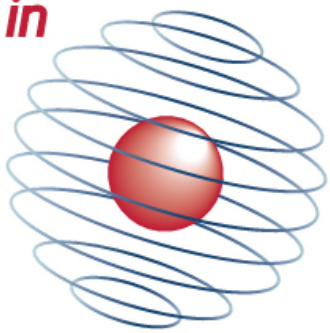
CENTRE *for* DOCTORAL TRAINING *in*
CYBER SECURITY



CDT Technical Paper

21/15

Privacy-awareness in Blockchain-based PKI

Louise Axon

Privacy-awareness in Blockchain-based PKI

Louise Axon, University of Oxford
louise.axon@cs.ox.ac.uk

Abstract—Conventional public key infrastructure (PKI) designs are not optimal and contain security flaws; there is much work underway in improving PKI. The properties given by the Bitcoin blockchain and its derivatives are a natural solution to some of the problems with PKI - in particular, certificate transparency and elimination of single points of failure. Recently-proposed blockchain PKI designs are built as public ledgers linking identity with public key, giving no provision of privacy. We consider the suitability of a blockchain-based PKI for contexts in which PKI is required, but in which linking of identity with public key is undesirable; specifically, we show that blockchain can be used to construct a privacy-aware PKI while simultaneously eliminating some of the problems encountered in conventional PKI.

1 Introduction

There is much current work in securing public key infrastructure (PKI). High-profile events such as the hacking of Dutch certificate authority (CA) DigiNotar in 2011 have further encouraged work in the improvement of PKI security. An emerging solution for building more secure PKIs is blockchain - first introduced in 2008 as the technology underlying Satoshi Nakamoto's cryptocurrency Bitcoin [1]. Proposed blockchain solutions provide desirable security properties, but do not give appropriate privacy guarantees and are as such unsuitable for many PKI applications; in this work, we give a construction for a blockchain-based PKI which provides varying levels of privacy.

The conventional approach to PKI is a centralised one that uses certificate authorities (CAs); web-of-trust (WoT) models such a PGP, and simple public key infrastructure (SPKI) have also advanced as options for PKI in recent years. In the CA system, CAs are trusted entities, who issue a signed certificate to an entity on request, certifying ownership of a public key by said entity. WoT systems are based on networks of trust: members of the network establish trust in others by verifying that those others are trusted by at least one already-trusted entity; that their certificate is signed by some entity in whom the verifier has previously established trust. These approaches have flaws in terms of security: in summary, CAs are single points of failure and the system can be subverted, partly due to a lack of sufficient transparency in the issuance of certificates; web-of-trust systems have such a high barrier to entry that it is difficult to join the network without being previously trusted.

Blockchain technology is fitting for the requirements of PKI, and holds advantages over the conventional approaches to PKI: in a decentralised blockchain-based PKI, the single points of failure represented by CAs in the conventional PKI stucture are eliminated, and a ledger of PKI events is published that is reliable as long as the majority of contributors are "honest parties"; we further detail blockchain's provision of these properties in Section 4. However, the way in which blockchain functions - as a *public ledger* in which actions are transparent - means that it does not naturally provide any privacy. The provision of privacy, to prevent tracing and linking of identities and their actions, is of vital importance in PKI - in particular in emerging applications of PKI such as the Internet of Things, mobile networks, smartcards and vehicular networks (these applications are discussed in Section 4) - and it is therefore important to establish that blockchain-based PKI can provide this privacy.

Some work has been published recently in the construction of blockchain PKIs and the evaluation of their security. Certcoin[2] is as yet the only fully-detailed architecture for a blockchain-based PKI. In Certcoin, entities establish links with their public keys by posting identity and public key as a pair of values to the blockchain. Each public key can therefore be linked with its corresponding identity by all entities who can view the blockchain and the actions of entities can be traced in this way. Privacy is not provided by the Certcoin architecture; the

privacy-aware applications described above require a PKI with a level of user privacy that cannot be provided by PKIs in which the actions of users within the PKI - linking of identities with public keys, and changes in public keys - are visible to others. Adaptations must be made to give a PKI more suitable for applications requiring privacy, and this is the basis of our proposal, given in Section 5.

Our contribution is a decentralised PKI with unlinkable short-term key updates and user-controlled disclosure - the *user* of a public key controls the disclosure of his identity and of his previously-used public keys. It is an extension of the blockchain PKI to fit situations in which privacy-awareness is required; we show that a secure PKI can be built on the blockchain that does not require public linking of identity to public key, and we outline an architecture for this.

In Sections 2 and 3 we give relevant background in the requirements of and common approaches to PKI, the blockchain architecture, blockchain-based PKI and the Certcoin [2] proposal for a blockchain-based PKI. In Section 4 we derive an appropriate notion of privacy - user-controlled disclosure - by examining the required properties of, and detailing use cases for, privacy-awareness in blockchain-based PKI. We present our proposal for a privacy-aware blockchain-based PKI in Section 5, and give our analysis of this proposal in Section 6. We detail related work in Section 7.

2 Background: Preliminaries

2.1 PKI and privacy-aware PKI

PKI is a system by which public keys are managed. Public key cryptography requires entities to have a key pair comprising a public and a secret key. The PKI provides a record and authentication of the link between a public key and its owner, usually based on certificates verifying the ownership of a public key (and the corresponding secret key) by some entity. A PKI must support certain functions: the registration and update of public keys, as well as appropriate mechanisms - usually revocation or backup - for coping with key compromise or loss.

There are generally two approaches to PKI. The first, and most common, approach, is CA-based PKI - specifically, the X.509 standard - in which the CA is a trusted third party, and must be trusted by all parties involved in any transaction. Certificate authorities issue certificates authenticating the link between a public key and its owner. In order to "trust" a CA, devices accept a root certificate for that CA into its store. From this root CA stems a hierarchical certificate chain, in which any certificates signed by a trusted certificate are also trusted. A signature on a certificate, linking an identity with a public key, by a CA trusted by a user should authenticate to that user the ownership of the public key by the identity.

The Web-of-Trust (WoT) approach to PKI was proposed by Phil Zimmerman in 1992 in the manual for his Pretty Good Privacy (PGP) program for encryption and decryption. Unlike the CA-based PKI, in the WoT approach trust is decentralised: a key is trusted by an entity if it is verified by another trusted entity; as Zimmerman explains: "...everyone will gradually accumulate and distribute with their key a collection of certifying signatures from other people, with the expectation that anyone receiving it will trust at least one or two of the signatures."

There are a number of disadvantages to these approaches to PKI; we discuss these further in Section 3, and show in Section 4 that decentralising PKI using the blockchain can circumvent some of these weaknesses.

By privacy-aware PKI, we describe PKI that is built to provide some level of privacy - where we consider privacy to be the ability of the user to control their disclosure of information - greater than that achieved by a conventional PKI, in which identities are publicly linked with their public keys. This level may be complete anonymity, or may be some lower level of privacy - in particular, in this work, we focus also on the privacy that can be achieved by remaining anonymous to all network members except a small "neighbour" subset. Different applications require different levels of privacy, and we consider the privacy required for a set of use cases in Section 4.

2.2 Blockchain

The blockchain was first introduced as the technology underlying Satoshi Nakamoto's cryptocurrency Bitcoin [1] in 2008. The blockchain is a public ledger to which events are posted and verified by network members before being confirmed - "mined" - in an incentivised system in which members must compete to complete some proof-of-work - usually a cryptographic challenge.

In Figure 1 we give a basic model of the Bitcoin blockchain, taken from the Bitcoin Developer Guide [3]. The contents of previously-mined blocks are hashed and contained within the following block, so

that the details of a past transaction can only be altered through altering every block mined following that transaction. This creates a reliable transaction record which can only be altered by a mining power that constitutes a majority of the mining power of the entire network.

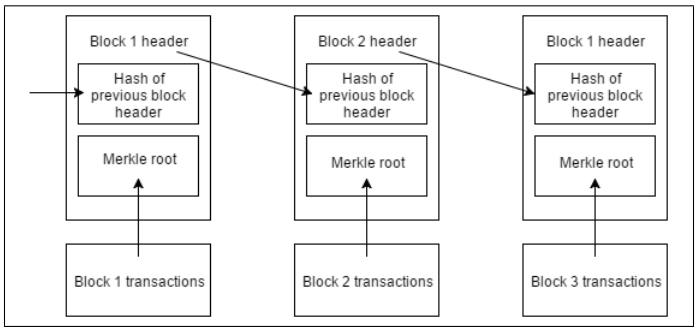


Figure 1: Bitcoin blockchain architecture

The merkle root is a hash of transactions per block, much smaller in size than the entire record of transactions per block, and can be used to securely verify transactions. The merkle roots eliminate the requirement to download the entire blockchain for verification of transactions.

The blockchain has a unique combination of properties that make it suitable for a number of applications - aside from PKI, proposed and existing applications include smart contracts, reputation systems, and interaction between devices for the Internet of Things. In particular, it is decentralised (no trusted third party is involved in its operation), and events recorded in the past cannot be altered (without control of the majority of the network's mining power - for further explanation of this we refer the reader to [1]).

Following this introduction of the Bitcoin blockchain, a number of alternative blockchains have been proposed; of particular importance for this work is the Namecoin blockchain [4]. Namecoin is a cryptocurrency and works as a decentralised domain name server (DNS). Certcoin[2] is built on the Namecoin blockchain, as is our proposed PKI in Section 5. Namecoin is based on the Bitcoin blockchain but has a key difference that makes it suitable for wider applications: Namecoin's blockchain can store data. Mining in Namecoin is done by the same proof-of-work algorithm as in Bitcoin.

Having established these basic details for Namecoin, we do not focus on them for the rest of this work; for the building of our PKI proposal, it is important only that the Namecoin blockchain is a public ledger to which transactions can be posted, and on which past actions cannot be reversed without subversion (by a dishonest majority) of the blockchain.

3 Background: Blockchain PKI

3.1 Why build PKI on the Blockchain?

The basic architecture of a blockchain PKI is as follows. The blockchain functions, as in the confirmation of Bitcoin transactions, to register the linking of a public key to an identity, and to confirm this link for subsequent actions. PKI functions - registration, update, revocation - are performed by posting the identity and public key with a required action to the blockchain.

We evaluate the blockchain-based approach to constructing PKI and show its merits, before continuing to our privacy-specific proposal. Through comparing the blockchain-based approach to PKI with the CA- and WoT-based approaches, we identify the advantages of blockchain PKI that should be retained, and this informs our proposal in Section 5.

Blockchain is well-suited to PKI, and provides desirable properties: certificate transparency and revocation, elimination of central points of failure, and a reliable transaction record. We begin by comparing CA-based PKI with blockchain PKI; in particular, we detail the problems with CA-based PKI, and show that using blockchain can naturally alleviate some of these problems. In order to set out the problems with CA-based PKI, we draw on those detailed in Section 7; of particular importance those detailed in [5].

CA-based PKI contains numerous single points-of-failure. The centralised trust model of CA-based PKI means that the security of the system is based on single points: the CAs themselves. Furthermore, the structure of the certificate chain means that if one single CA in the chain is rogue or subverted, this can compromise the security of the whole chain. Building PKI on the blockchain removes the potential points of failure that each CA represents.

In CA-based PKI, revocation is generally handled through Certificate Revocation Lists (CRLs) - lists of certificates that have been revoked. It is well-established that handling revocation through such processes can be costly in terms of time. Revocation for blockchain-based PKI can be solved efficiently using distributed hash tables.

Google's Certificate Transparency Project [6] looks to aid detection of rogue CAs and certificates. Certificate Transparency (CT) is given by append-only public logging of digital certificate informa-

tion, where users can view the log and hence verify the digital certificates they use - this is transparency. Using the blockchain for PKI gives a natural provision of the transparency property that Google look to achieve in this project - the blockchain functions as a public log, and is append only (past logs cannot be changed unless the blockchain is subverted by a dishonest network majority) - while simultaneously eliminating the need for trusted CAs present in CT[6].

In WoT-based PKI, network members are "trusted" if their "trustworthiness" is attested by some other "trusted" network member. In order to build these "trust" relationships, new network members must somehow gain the trust of some network members, since it is based on this trust that their "trustworthiness" can be attested to others. This need to establish trust means that there is a high barrier to entry for a new member in a web-of-trust; particularly, the amount of work required to gain a web by which to prove "trustworthiness" to a usefully large proportion of the network is high.

Blockchain-based PKI does not have this requirement for a web of attesting, "trusting" members for each entity, and so this work required before performing as a network member is removed.

We conclude that decentralising PKI gives desirable security properties that are not achieved by conventional approaches to PKI. In the remainder of this paper, we turn our attention to the specific problem of a *privacy-aware* decentralised PKI.

3.2 Certcoin: a blockchain-based PKI

We give the outline of the Certcoin architecture, given in [2], for a blockchain PKI that is not privacy-aware. This is presented in the same format as we later present our architecture for a privacy-aware blockchain PKI, in order to give clarity and aid comparison. In [2], three versions of the PKI are given, in two of which efficiency is improved through the incorporation of cryptographic accumulators and distributed hash tables. Here, we focus on **version 0**, the most basic version given before efficiency is improved (in **version 1** and **version 2**, detailed in Section 6). The notation in this presentation of Certcoin is adapted somewhat from the original Certcoin proposal ([2] and [7]) to enable better comparison with our proposal later; however the protocol presented is unaffected by this slight change in notation.

High-level description and diagram

Certcoin is built on the Namecoin[4] blockchain, details of which are given in Section 2. Certcoin functions as follows: the blockchain forms a ledger to which identity and public key are posted in pairs, along with the action (registration, update, revocation), and processed through verification and mining by the network. A broad picture of this architecture is given in Figure 2.

Setup/ initial registration

- sig is a digital signature algorithm
- ver is a verification algorithm that evaluates to 0 or 1

Offline and online key generation

Identity owner generates[1]:

- an online public and secret key pair (pkn_0, skn_0), and
- an offline public and secret key pair (pkf_0, skf_0)

Key registration

Identity owner posts:

- (**id**, register, online, values=(pkn_0, σ_{n0})), where σ_{n0}=sig(skn_0, **id**) and is given to demonstrate ownership of the online secret key skn_0 corresponding to public key pkn_0, and
- (**id**, register, offline, values=(pkf_0, σ_{f0})), where σ_{f0}=sig(skf_0, **id**) and is given to demonstrate ownership of the offline secret key skf_0 corresponding to public key pkf_0.

Verification

It must be verified that:

- **id** has not been registered previously
- ver(pkn_0, σ_n, **id**)=1
- ver(pkf_0, σ_f, **id**)=1

Key updates

Either the online or offline key pair may be updated: either old online public key pkn_{n-1} can be updated to new online public key pkn_n, or old offline public key pkf_{n-1} can be updated to new offline public key pkf_n.

1. It is assumed that the user can generate these values locally

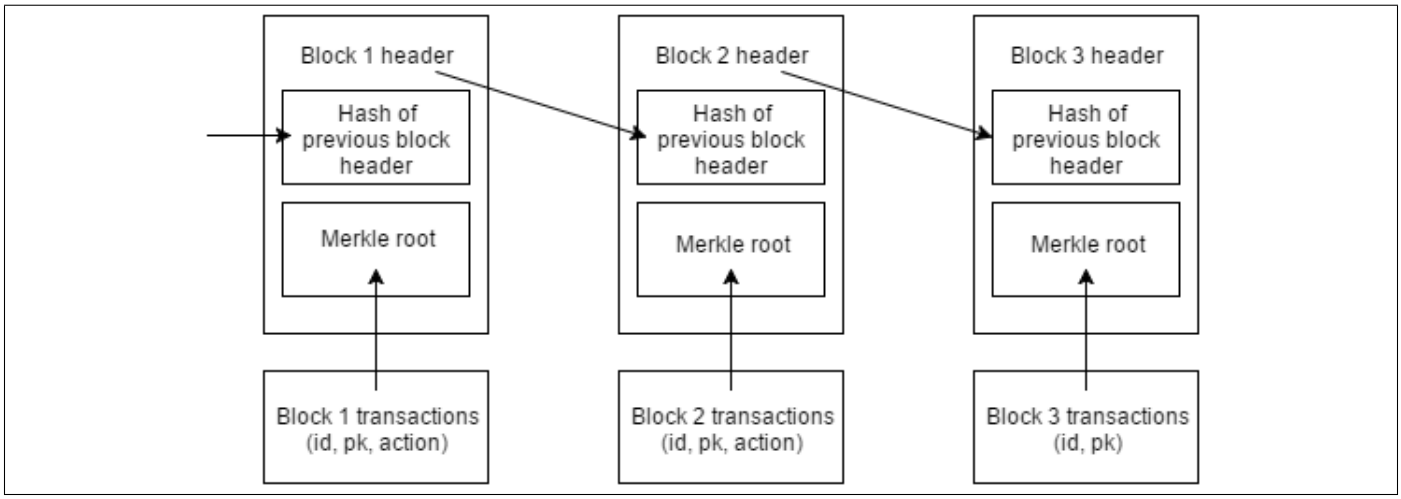


Figure 2: Blockchain PKI architecture

Offline and online key generation

Identity owner generates:

- an online public and secret key pair (pkn_n, skn_n), or
- an offline public and secret key pair (pkf_n, skf_n), to be stored offline

Update registration

Identity owner posts (for either online or offline key update, respectively:

- (**id**, update, online, values=(pkn_{n-1}, pkn_n, σ_{nn1}), σ_{nn2}, aux), where σ_{nn1}=sig(skn_{n-1}, (**id**, pkn_n) - the identity and new public key, signed by the old secret key - and is given to demonstrate ownership of the old secret key skn_{n-1} corresponding to public key pkn_{n-1}; σ_{nn2}= sig(skn_n, **id**) - the identity signed by the new secret key - and is given to demonstrate ownership of the new secret key skn_n corresponding to new public key pkn_n; aux is an auxiliary message that may be required in case of key compromise (we detail this no further here), or
- (**id**, update, offline, values=(pkf_{n-1}, pkf_n, σ_{fn1}), σ_{fn2}, aux), where σ_{fn1}=sig(skf_{n-1}, (**id**, pkf_n) - the identity and new public key, signed by the old secret key - and is given to demonstrate ownership of the old secret key skf_{n-1} corresponding to public key pkf_{n-1}; σ_{fn2}= sig(skn_f, **id**) - the identity signed by the new secret key - and is given to demonstrate ownership of the new secret key skf_n corresponding to new public key pkf_n

Update verification

It must be verified that:

- pk_{n-1} corresponds to **id**
- ver(pk_{n-1}, σ_{n1}, (**id**, pk_n))=1
- ver(pk_n, σ_{n2}, **id**)=1

Recovery, revocation and expiration

Key recovery is enabled through social backup: Certcoin requires the setup of a recovery system - the secret key for an entity must be secret shared (by Shamir secret sharing [8], for example) between at least three trusted "friends", and reconstructed with a threshold of at least two.

The revocation process differs depending on whether the adversary only accesses, or also steals, the secret key, and on whether it is just the online secret key, or both keys that are accessed or stolen. Briefly, the revocation works as follows. If only the online secret key is lost or stolen, then the ownership of the offline secret key means the true owner still has power over the adversary to prove his ownership of the online secret key. If the adversary gains access to, but does not steal, both online and offline secret keys, then the adversary can no longer be distinguished from the true owner, so the owner can use both keys (still having access to them) to invalidate their present and future use. If both online and offline key are stolen, however, then there is no revocation process within Certcoin; the keys are then controlled by the adversary.

It is stated that Certcoin public keys expire after a given lifetime, verified according to a timestamp.

As explained in Section 1, Certcoin is not suitable for uses that require some level of privacy, because it is built as a ledger in which identity is

posted publicly with public key. This means that all actions made with any public key can be traced to a particular identity by any entity who views the ledger. Furthermore, all key updates are linked to the previous public key of the updating entity: it is stated in [2] that an "update transaction will only be processed if the signature verifies with the old public key" - in order to have a key update processed, any entity must reveal his reveal his previous public key.

4 Privacy Requirements

4.1 PKI Use Cases

We examine the situation that a PKI is required - more specifically, a network in which entities may communicate securely using public keys is required - but the linking of public keys with identities is undesirable. Such situations arise particularly where it is required that participating entities' actions cannot be tracked by their use of public keys: some applications in the Internet of Things, in ad-hoc networks, vehicular networks, and online anonymous forums, networks similar to Tor. We begin by detailing a set of such cases in which privacy-aware PKI is required, and assess the type of privacy required in each case. It should be noted that, aside from these specific use cases, the provision of privacy is a concern for PKI in general - to prevent surveillance and tracking or profiling of users' actions - and that the provision of privacy in conventional PKI is often insufficient (Brand in particular argues the requirement for the building of privacy into PKI in [9]).

- **Ubiquitous computing:** In ubiquitous computing, a user's computer may exist in many different forms: interactions with a computing system may be made through multiple devices (laptop, smartphone, tablet, for example). This has heavy privacy implications - the linking of a user's actions through multiple devices can enable linking of the devices and tracing of the user's actions and location. There is the requirement for a level of privacy in which by default the user's identity is not linkable across devices, and his public key is not repeatedly used across devices.
- **Internet-of-Things:** This is similar, and an extension to, the ubiquitous computing environment given above. Again, a single user may act on multiple devices (smart TV, fridge, for example) and the linking of the identity using the devices, or of the public key corresponding to an identity being used across devices, is a privacy concern.
- **Vehicular networks:** Vehicular networks require PKI for secure inter-vehicular communications, but it is necessary that the use of public keys by vehicles does not enable remote tracking of that vehicle and its actions. It is therefore required that the identity corresponding to any particular public key is not publicly disclosed, and that public keys are unlinkable at updates by any party who should not be able to track the vehicle.
- **Anonymous forums and networks:** Online forums and networks in which users require total anonymity require a system - a PKI - in which users can verify that they belong to the network in question, but need disclose no further information pertaining to their identity or linking the actions they perform on that network. The public keys of an entity in such a case should not be frequently changed and unlinkable, and the identity not disclosed.
- **Smart cards:** Smart cards are used for multiple purposes - to authenticate payments, prove credentials, prove identity. Since any single smartcard may be used in multiple locations and for multiple purposes, it is important that its use cannot be tracked by repeated use of public key, or by disclosure of identity (directly, or through disclosure of a public key publicly linked with an identity) at each use.

4.2 Derived notions of privacy: user-controlled disclosure and "neighbour group anonymity"

The range of required levels of privacy identified above point to a requirement for adaptability in the privacy provision of the PKI. We address this requirement in two ways: firstly, we build into our PKI a user-controlled identity disclosure mechanism whereby each user chooses whether and when to disclose their identities or past public keys; secondly, we begin our proposal in Section 5 with a PKI in which users have "total anonymity", and then show how "neighbour" groups can be used to give a different type of privacy - "neighbour group anonymity".

By "total anonymity", we mean that for each entity **E**, no other entity can link the public keys of

E to any other of **E**'s public keys, or to his identity. "Neighbour group anonymity" describes the case that the actions of an entity are identifiable within a trusted "neighbour" group, but remain anonymous to the rest of the network.

For "total anonymity, we give in Figure 3 the key-posting for a group of anonymous entities on a network, who wish to establish secure, yet anonymous, communications with other network members. In Figure 3, network members essentially pool public keys in view of the rest of the network, so that messages can be sent to non-specific network members, or broadcast to the network as a whole, without knowledge of the identities corresponding to the public keys. This is the structure of a network giving our highest level of privacy - "total anonymity".

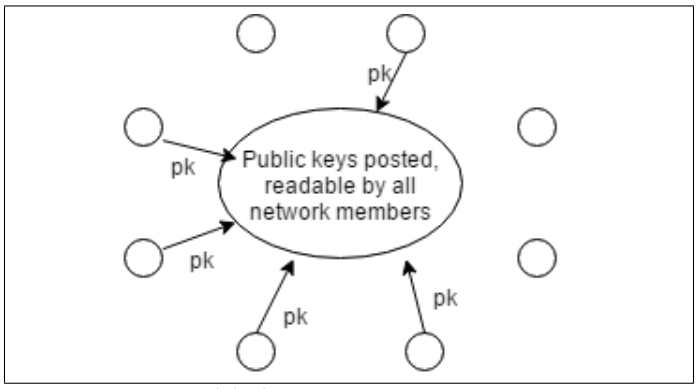


Figure 3: Establishing a secure communications channel using public key cryptography: anonymous network members

To illustrate our notion of "neighbour group anonymity" we provide Figure 4. In this diagram, nodes 1, 2 and 3 form a neighbour group on a network of many nodes. This means that nodes 1, 2 and 3 have some level of trust and each will disclose some identiying, or key-linking, information at updates to the other two members of their neighbour group, to be verified by these two members and the correctness of the update then attested by these two members to the rest of the network.

In our proposal and analysis, we find that there is a trade-off between PKI security and the privacy level the PKI is capable of providing. Specifically, we find that with the relinquishing of some anonymity from the "total anonymity" level to some "neighbour group" level comes an increase in the efficacy of functionalities of the PKI, in particular with respect to tracing misbehaviour, and hence an improvement in its security in some respects. Generally, we find that security is inadequate in the case of total anonymity, and that some level

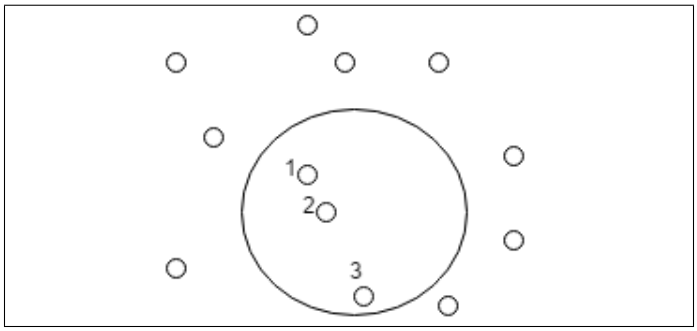


Figure 4: Neighbour group anonymity

of neighbour group anonymity is more suitable in most cases. In Section 5 we build a totally anonymous PKI as a base on which neighbour group anonymity is then built. The security and privacy provided by this is further discussed in Section 6.

4.3 Specific privacy requirements

Given the above description of the PKI and its use cases, we list the requirements for it. These requirements are adapted from those in [2], in which certain necessary functionalities are given for the PKI. In [2] the PKI requirements are given as follows (the following cited from [2]):

1) Registering an identity with a corresponding public key
2) Updating the public key corresponding to a previously-registered identity
3) Looking up a public key corresponding to a given identity
4) Verifying that a given public key corresponds to a given identity, perhaps in a way more efficient than performing a lookup, and
5) Revoking the public key corresponding to an identity.

The requirements for our *privacy-aware* PKI are somewhat different to these. Our requirements are derived from the design problem: retaining the necessary PKI functionalities while enabling the most extreme privacy level identified - "total anonymity". The required functionalities derived for our PKI are as follows:

1) Registering an identity with a corresponding public key
2) Updating the public key corresponding to a previously-registered identity
3) Looking up a public key valid on the network
4) Verifying that a public key is valid on the network, and
5) Revoking a public key from the network

We justify these requirements:

1) It is necessary, for a PKI, that an identity be in some way linked to a public key. If this link is not present at some point, then we have not a PKI but a list of keys. (Anonymity begins at the first short-term key change, which can be initiated immediately following this initial step).
2) It is necessary that an entity updating a public key has some identity established on the network.
3) For communications, entities on the network must be able to look up other public keys on the network. Our requirement here has a different meaning to the corresponding requirement in [2]: while Certcoin enables the lookup of the public key corresponding to a particular identity, we describe here only the lookup of *some* public key on the network.
4) It is required that all public keys can be verified as belonging to the network (again we differ from [2] here in that the verification of a public key against its identity is not required).
5) It is required that public keys can be revoked from the network, without necessarily having knowledge of the corresponding identity.

Our proposal, and the architecture set out below, takes root in the observation that in order to meet these requirements for the privacy-aware PKI, identity and public key should not be publicly linked. Our proposal avoids the public linking of public key with identity or with previous public keys for any entity.

For tracing and revocation purposes, in case of misbehaving entities on the network in particular, it is desirable that information linking public keys and identities is not *completely* absent from the system, but rather that access to it is controlled by the system in a decentralised manner; particularly, that access to information can be given following the consent of a majority of the network or of the user who owns the information.

5 Proposal: Blockchain PKI with Privacy-Awareness

High-level description and diagram

Our proposal is based on the separation of the identity value from a series of short-term public keys posted to a publicly-verified ledger - the blockchain. Rather than posting both identity and public key to the blockchain, and reposting identity at updates, such that any public key can be linked to its owner at any time as in Certcoin, we propose a PKI in which once an identity **id** is established, its key updates are anonymous unless the owner of **id** chooses to reveal them (to prove key ownership in case of compromise, or if forced to for reasons of liability, for example). This hidden linking is enabled by the offline secret keys, which are part of the function by which updated online public keys are generated, and so create a link between online public keys and initially-registered identity. This link can be proved, but is not revealed at key updates, and so the unlinkability of key updates is preserved unless it is required that they be revealed (by the key owner, or by some authority, for example, in cases related to liability or misbehaviour).

We illustrate the offline key linking process of our PKI in Figure 5 for an entity **E**. In Figure 5, notation for the key types is given as later in the detailed architecture. pkf and skf are the offline public and secret keys respectively, while pkn and skn are the online keys. As illustrated, the new online public key at each update is computed as a function of the previous online public key and the offline secret key. While each short-term online public key pkn posted is publicly unlinkable to the last, **E** retains a linking record of his offline keys (used in the online key update function) by which he can prove his ownership of past online keys and prove the link between his public key and identity. In this way, a chain of online public keys is created that is verifiable right back to the initial identity establishment, and which the owner can choose to reveal.

As explained in Section 4, our proposal is designed to enable two levels of anonymity: total anonymity, and anonymity to the neighbour group level. In this initial presentation of the architecture, we detail the system for providing total anonymity. Following this, we describe the process for achieving neighbour group anonymity and with it, as discussed, some improved PKI functionality and security. This trade-off between security and privacy was introduced in Section 4 and is further discussed in Section 6.

As in Certcoin [2], we build our PKI on the Namecoin blockchain [4] detailed in Section 2; for clarity, and to aid with comparison, we present the stages of our PKI in the same structure as the presentation of Certcoin given in Section 3.

The key difference between our proposal and Certcoin is in the key update process. While similar to Certcoin in the initial registration process - with the public establishment of an identity linked with a public key - our proposal has a key update process

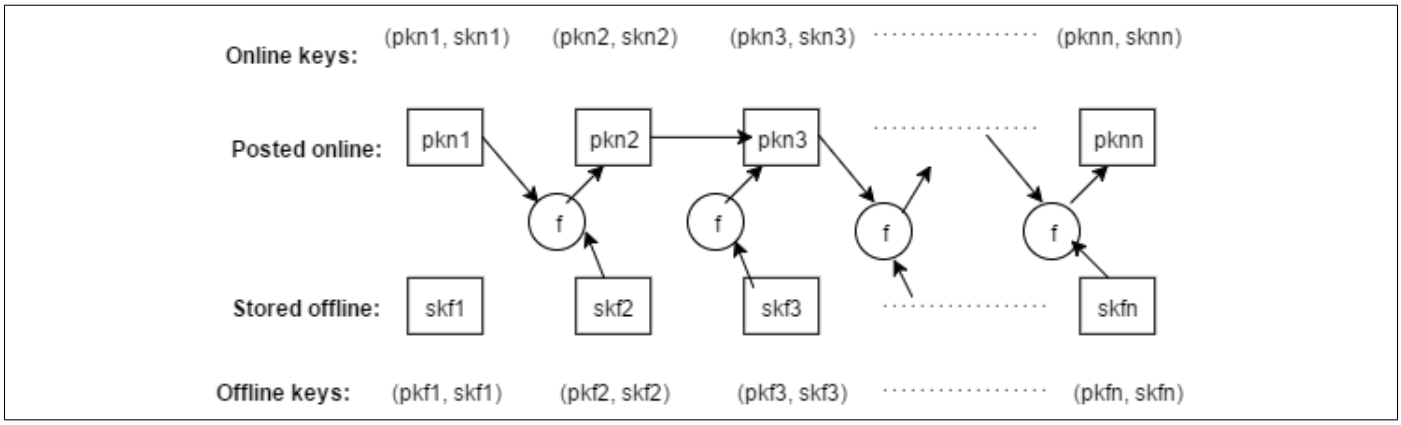


Figure 5: PKI architecture

designed to give two things: no public link between identity and updated public key, and no public link between previous public key and updated public key. Instead of this public linking, a hidden link is created between key updates that can be traced back to an initially-posted identity, as described. As such, our key update process enables the user to update his public key anonymously. Some authority is required in some networks, so we decentralise the control over authorities' access to information to require the consent of a majority of the network (or of the user who owns the information). We first describe the initial, public registration process, very similar to that in [2], and then describe the key update process.

Setup/ initial registration

- sig is a digital signature algorithm
- ver is a verification algorithm that evaluates to 0 or 1

Offline and online key generation

Identity owner **E** generates[2]:

- an online public and secret key pair (pkn_0, skn_0)
- an offline public and secret key pair (pkf_0, skf_0)
- a nonce N_0

Key registration

E posts:

- (**id**, register, online, T_0, values=(pkn_0, σ_i), where T_0 is a timestamp, σ_i is the initial value signature sig(skn_0, **id**) - the identity **id** signed with online private key skn_0 (this proves **E**'s ownership of the online secret key skn_0 corresponding to online public key pkn_0).

This is entity **E**'s first establishment of his identity **id** in the PKI. The remaining information generated (skn_0, pkf_0, skf_0) is retained by **E**.

Verification

It must be verified that:

- **id** has not been registered previously
- pkn_0 has not been registered previously
- ver(**id**, pkn_0, σ_1)=1

Unlike [2], we do not require verification of **E**'s ownership of the offline key pair in this initial stage. This is therefore omitted from the algorithm.

Key updates

The offline key pair at each update is randomly generated, as in the initial registration stage. Two nonces N_n and R_n are also randomly generated for the n^{th} update.

In order to update a key, entities must prove that they are network members - that they already have a current online public key registered in the PKI; this stage is given in the verification below and its efficiency evaluated. This means that only already-established network members may attempt to attack other network members by revoking their public key (this attack is further discussed in Section 6).

For the key updates, the online key pair is generated as a function of the previous online key pair, and the newly-generated offline key pair and nonce - so for the n^{th} online key pair, pkn_n=f_1(pkn_{n-1}, skf_n, N_n) and skn_n=f_2(skn_{n-1}, skf_n, N_n), where f_1

2. As in [2], we do not detail the generation of values in this work, but assume that **E** generates these values locally

and f_2 are the modular functions described below. In summary, the reason for this is create a verifiable chain of updates; to enable verification for an entity **E**, if required, that **E** did (or did not) make a chain of public key updates - **E** can choose to prove this by proving that his chain of online public keys were indeed generated using his offline keys (which should only be known by **E**). In case of compromise of these offline keys, an adversary could impersonate **E**, and we therefore require the social backup mechanism described in the revocation section.

Offline key generation

E generates a new offline public/secret key pair, for the n^{th} update (pkf_n, skf_n).

Online key generation

For the n^{th} online key pair, **E** calculates:

- $pkn_n = \mathbf{f}_1(pkn_{n-1}, pkf_n, N_n) = pkn_{n-1} \times skf_n(modN_n)$, and
- $skn_n = \mathbf{f}_2(skn_{n-1}, skf_n, N_n) = skn_{n-1}/skf_n(modN_n)$.

(Note: this is cryptosystem-specific - for an elliptic curve-based cryptosystem, for example, the above values would need to be constructed differently)

Update registration

E posts the tuple (register, online, T_n, values=(pkn_n, R_n, σ_{rn-1}, σ_{rn})) to the network, where T_n is a timestamp, σ_{rn-1} is the signature sig(skn_{n-1}, R_n) (this is used to verify that **E** already owns a public key - so has an established identity in the network - in step 3 below), and σ_{rn} is the signature sig(skn_n, R_n) (this proves **E**'s ownership of the online secret key skn_n corresponding to online public key pkn_n).

Network member update verification

1) number of **id**s=number of public keys (for the network as a whole - efficiency of this discussed in Section 6).
2) pkn_n has not been registered previously
3) ver(R_{n-1},pk, σ_{rn-1})=1 (where pk is some public key currently registered as in use on the network - so verifiers check through those public keys currently in use on the network, and check that one current online public key in the PKI satisfies this verification. This step verifies that submitter already had a registered public key previously - it is adopted here as an alternative to checking that the submitter has a registered *identity*, since identity is not given at updates. The efficiency of this step is again discussed in Section 6)
4) ver(R_n, pkn_n, σ_{rn})=1

Step 3 effectively means that *only network members* - those with public keys already registered to their own identity - may attack another network member by registering a new public key to their identity. This reduces the base of adversaries to network members only. If such an attack were to occur, and an adversary (and network member, necessarily) **A** were to register a new public key to identity **E**, then **E** could revert to his previous public key by proving his ownership of it using his offline keys. This is further detailed in the revocation section.

It is important that the verification in step 3 returns only 1 or 0, and does not return the value of pk found; otherwise key changes become linkable.

For tracing reasons in case of misbehaviour, as detailed in Section 6, at the point of updating his online public key each entity should share his previous secret key between a majority of network members using a secret sharing scheme. This creates a social backup mechanism for dealing with key loss.

Recovery, revocation and tracing

Recovery

If an online secret key is lost, it can be reconstructed using the offline secret keys, since online secret keys are a function of offline secret keys and moduli, and previous online public keys (for this reason, online secret keys should be stored offline once changed).

If an offline secret key is lost, there is no clear method of retrieving it. One option is for the owner to invalidate his online public key by signing a statement against it using his online secret key, and then to begin again from the registration stage, re-establishing his identity against a new online public key and beginning the key chain again. Another option is social backup: an owner may choose to secret share his offline secret keys with a few trusted "friends", such that in the case of offline secret key loss, the key may be reconstructed from their shares, using Shamir secret sharing [8] or similar. The loss of an offline secret key means that the owner no longer has the power to prove his identity if required, and so the key must be retrieved or the identity re-established in these ways.

If both secret keys are lost, then an owner may retrieve his offline secret key through social backup, as described above, and then use his offline secret key to prove the link between his current online

public key and his identity. This verification can serve as a replacement for the current online secret key, which can then be invalidated, and the identity can be re-established in the PKI, in the registration phase, against a new online public key.

Revocation and tracing identities

It may be required for some applications, vehicular networks for example, that some authorities have the power to trace actions back to their owner - particular in liability-related cases. Here, the authorities can demand that a targeted public key reveal his offline secret keys, present and historic, and his moduli N, again, present and historic. Since all online public keys of an identity are formed as a function of the previous online public key, and new offline secret key and modulus N, previous public keys can be reconstructed from this information, and the previous public keys traced back until the original posting with identity. In this way, identity can be reliably retrieved. Furthermore, this procedure is secure against identity spoofing, since to post another identity in the network can only be done with possession of all of their offline secret keys. Without this knowledge, a network member is unable to pose as any network member other than himself.

Should it be required that a key is revoked from the network, an identity can revoke his online public key by signing a statement, using his online secret key, that invalidates the public key. Again, this could be ordered by authorities in some cases.

In case of key compromise, an owner may need to take one of several actions. As in Certcoin, this depends on which keys have been compromised and to what extent: online secret key or both online secret key and offline secret key; accessed by the adversary or stolen. We detail each of these four cases.

- **Case 1: online secret key only, accessed:** The owner may use his online secret key to invalidate the online public key, before re-establishing his identity against a new online public key.
- **Case 2: online and offline secret keys, accessed:** As in Case 1, the owner may use his online secret key to invalidate the online public key, before re-establishing his identity against a new online public key. Note that the adversary, in possession of the online secret key, could perform the same action, invalidating the online public key. However this is not a serious problem: an adversary's capabilities with respect to the identity would then end here, since the identity becomes void.
- **Case 3: online secret key only, stolen:** The owner may use his offline secret keys to prove his ownership of the current online public key, and by this verification, authenticate an invalidation of that public key, before re-establishing his identity against a new online public key.
- **Case 4: online and offline secret keys, stolen:** The owner may reconstruct his offline secret keys through the social backup described, and use this to prove ownership of the current public key back to identity (starting from the identity posting with the first public key, and proving ownership of the offline secret key involved in all changes). Then the owner regains possession of both online secret key and offline secret key, and the case is reduced to Case 2 - that of access.

In revocation, we again perform the verification required in update that the revoker is a network member; as discussed, this can also be done through "neighbour" groups in lower privacy applications.

As in Certcoin, if unchanged, keys should expire after a certain length of time, judged by the timestamp.

"Neighbour" groups

Entities on the network may form groups of "trusted" members. In reality, this trust may be based on social standing/"friendship" - in forums, for example - or nearest neighbours in mobile networks - vehicular networks, for example. The idea is that within its group, an entity does not remain anonymous, and so its other group members can check its actions and attest their correctness to the rest of the network. This improves the security of certain functions in the PKI: the correctness of the identity performing key updates and revocations can be checked. As discussed in the next subsection, the members of these neighbour groups can also perform simultaneous key updates, preserving unlinkability of keys towards the rest of the network by preventing linking of old and new keys at updates by time association.

These groups give a different type of privacy to a participating entity. While an entity **E**, if part of a neighbour group remains anonymous, and his keys unlinkable, towards all parties that are not part of

his neighbour group, he is not anonymous towards the other members of his neighbour group, and his keys can be linked by them. There are applications where this level of privacy is appropriate, and total anonymity is not required; particularly, vehicular and other ad-hoc networks, in which it is illogical to seek anonymity towards the group of entities that are physically *nearby*, or within line-of-sight, at any one time, especially given that the aim of unlinkable key updates is such applications is the prevention of tracking. In vehicular networks, unlinkability is not required for the group that is "nearby" an entity, and so this "nearby" group can form a temporary neighbour group, attesting the correctness of key updates (that they were performed by the owner of the previous public key) to the rest of the network.

Simultaneous updates

There is a problem with the updating to a new online public key, and immediate discarding of the old one: that if at any particular time only one network member makes an update, then the single discarded old key on the network can be linked with the new one posted by time association, so the transaction becomes linkable. There are two ways of circumventing this, which fit in with two different levels of privacy - total anonymity, and neighbour groups.

1) **Simultaneous updates** In systems that use neighbour groups, as described above, every member of the neighbour group can update his key whenever any single member needs to update. An entity **E** can inform his neighbour group that a change is required, at which each member of the neighbour group can simultaneously update. The privacy achieved here is at the neighbour group level - not total anonymity.
2) **Random time delay** In systems where total anonymity is required, a random time delay can be instigated from the time of posting a new online public key, to the discardment of the old one. In this time, both are valid public keys for the entity, until that time (at the end of the delay) when the old public key is discarded. This prevents the problem described, and so preserves unlinkability of key updates while also preserving total anonymity, since neighbour groups are not required.

6 Analysis

6.1 Attack types

An adversary might have some attack capabilities against our PKI aside from those gained from the stealing of or access to a party's secret keys, which are discussed as part of the proposal in Section 5.

Network members have far greater adversarial capabilities than non-network members: since the verification process at key updates involves only checking that the party involved is a network member (and some checks on the number of keys on the network, as given in Section 5), an adversarial network member may effectively "update" the public key of a targeted party to a new public key under the control of the adversary, by using his status as a verifiable network member to remove the targeted public key, and register a new public key on the network. Similarly, an adversarial network member may revoke the online public key of another member - the online public key for a network entity **E** could be revoked by some network member other than **E**, leaving **E** unable to communicate over the network until he retrieves and proves ownership of his removed key by the mechanisms described previously.

In the total anonymity case, without the presence of neighbour groups, it *is* possible for network members to cause disruption by temporarily changing or removing other members' keys in this way - revoking the target member's public key and then registering a new public key on the network. Our mechanism allows retrieval of prior keys and therefore identity, but does not prevent other members from actually changing the keys for the period of time until the true owner retrieves his previous key. Since network members are anonymous, such attacks cannot be *targeted* but can be made at random. Neighbour groups prevent an adversarial network member from attacking in this way, since the "neighbours" of a targeted party in this case would attest to the network that the change being made was not initiated by the correct identity, and the change would therefore not be processed.

In terms of the security of the blockchain architecture itself, there are some flaws. In particular, the Namecoin blockchain is susceptible to subversion by a dishonest majority (51%)- it relies on the majority of miners being honest parties. It is important to consider the likelihood of such a collusive majority, since the security of our PKI relies on the blockchain being unsubverted.

6.2 Identification of misbehaving users

Our PKI enables the identification, by some authority, of adversarial network members by two methods - one which requires a majority consensus, and one which requires the consent of the network member corresponding to the identity in question. The two methods below correspond to the "total anonymity" case without neighbour groups. In the "neighbour group anonymity" case, this tracing becomes easier.

Note that by "authority" we do not signify any authority prescribed to the PKI, but an authority involved in the situation in which the PKI is used, that may need to trace misbehaving users. This may be, for example, the police requiring identification of a misbehaving communications channel user in a vehicular network.

- **Majority consensus:** This is a network majority consensus to disclose the previous public key involved at any particular public key update. The verification of network member process for each update should be secured by a key secret shared between the nodes of the network (the entity updating should secret share his public key at this point between a majority of nodes). In case of misbehaving users, authorities can request access to (and may be granted access to by the majority of network members) the previous key at a particular update. If this majority consents, authorities may reconstruct the key from their shares and access the value of the public key used for that transaction. Publicly, though, and without this reconstructed key, the network member update verification process should return simply 1 or 0 (corresponding to network member or not). By tracing back through updates in this way, the authority can order the corresponding identity to revoke from the network. So, there is some authority, but the majority of the network controls its access, i.e. the access of the authority is decided in a decentralised way by a network majority.
- **Network member consent** Each network member can choose to prove his ownership of previous public keys, or the link between his public keys and identity. This means that authorities can send an order to a misbehaving public key to reveal this information.

6.3 The trade-off between security and privacy

The evaluation of attack types in the preceding subsection shows that we have traded off a certain amount of security in the PKI. In particular, that changes to an entity **E**'s key, or revocation of that key, may be made by some network entity other than **E**, with **E** then retaining the power to recover that key and regain control of his identity; despite this ability to regain control, it is clear that there is, in some cases, a window in which an adversarial network member may pose as **E**, or at least cause disruption. This trade-off is made to give a system for total anonymity. However, this attacker capability can be reduced, and security of the PKI therefore improved, by the neighbour group adjustment, for cases in which a slightly lower level of privacy i required. We examine this in the following comparison.

Comparison: total anonymity and neighbour group anonymity

To achieve total anonymity, we accept the fact that an identity can be temporarily disrupted, and only changed back by proving ownership of the chain. For total anonymity, and unlinkability of keys, this is intuitively the best we can do. To prove either ownership of the current public key, or prove the identity, at the time of update, would mean revealing either the current public key, or the identity - enabling either the linking of public keys or the knowledge of the identity to which the new public key belongs - this is not what we need here.

Different levels of privacy may be required, and with this different levels of security can be achieved. Our proposal to provide total anonymity comes as something of a trade-off with security: the verification of the online public key update extends only to the verification that it is made by a network member, and not that it is made by the owning identity for the online public key in question. In some use cases, however, total anonymity is not required; this is where "neighbour" groups who are in some way "trusted" by an entity **E** may view the identity of **E** and past keys at updates and may thus verify, for each key update, the performing identity, and attest to the rest of the network the correctness of the transaction. An example of such a "neighbour" setup is given for a specific case: ad-hoc networks. In a vehicular ad-hoc network, for example, we can form "neighbour" groups from those vehicles within a certain distance of one another. This enhances security and is ideal for vehicular networks

since preventing tracking by those vehicles who are nearby is futile.

6.4 Efficiency

[2] gives two further versions of Certcoin in which efficiency is improved. These can also be applied, in part, in our PKI. In **version 1** of Certcoin, cryptographic accumulators are introduced to improve the efficiency of testing membership for the set of network members. For further detail on this process we refer the reader to [2]. In **version 2** of Certcoin, distributed hash tables (DHTs) are introduced to improve the efficiency of public key lookup for a given identity. This lookup functionality is not part of our PKI, and so we do not require this construction.

We consider the efficiency of the network member update verification step in our PKI: the verification involves traversing all public keys in the network until one is met which verifies with the given verification algorithm; this confirms that the entity posting the updating public key to the network is a network member. The time involved in this traversing process can become part of the proof-of-work (blockchain proof-of-work described in Section 2), as long as it is smaller than the required proof-of-work for that particular network. As in Certcoin, this process could be made more efficient through the use of accumulators.

7 Related Work

7.1 Blockchain PKI

Some proposals have been made for blockchain-based PKI: [2] and [10] detail architectures for blockchain PKI.

The most complete architecture for a blockchain PKI is Certcoin, given in [2]. Here, blockchain technology is used to build a decentralised PKI that is fully-functioning, with the required PKI functionalities: registration, update, revocation of keys. The architecture is given in detail in Section 3, since our proposal is built on the proposal of [2], and for further detail we refer the reader to the original Certcoin paper [2]. In [2], the case is made that conventional PKIs do not guarantee *identity retention*: impersonation of already-registered identities is not guaranteed against. Certcoin aims to give better identity retention guarantees than conventional CA or WoT PKIs, where identity retention is the property that a user is prevented "from registering a public key under another user's already-registered identity." This is a property that the Google Certificate Transparency Project [6] detailed above looks to solve, and Certcoin provides an alternative solution to this problem.

In [10], the Nidaba decentralised PKI is proposed, with a focus on stable certificate registration price. It is built as a hash table of which certificates are elements: pairs of unique name and public key. The stability of certificate registration price is achieved through an adjustment of the base price for according to the performance of miners. The paper does not give details of the implementation, but provides some requirements - price stability, prevention of behindhand forking, and scalability - of blockchain PKI, and ideas for achieving them.

[11] gives details of and outlines processes for achieving the requirements of PKI built on the blockchain, drawing on the Certcoin architecture [2] in particular.

Related to the blockchain PKI are decentralised, blockchain-based domain name servers (DNS) that have been proposed and are currently in use. In particular, Namecoin [4], DNSchain [12]. DNSChain is a PKI built by the okTurtles Foundation [13] in which blockchain replaces X.509 certificates in a DNS and HTTP server. The details of NameCoin are given in Section 2, since it is on NameCoin that CertCoin is built. DNS gives only part of the function of a PKI - maintaining a record of domain names and their related IP addresses - and proposals for blockchain-based PKI have thus far been built as extensions of this decentralised DNS.

Proposals have been made in online forums of decentralising PKI using blockchain. Vitalik Buterin, the founder of decentralised platform Ethereum[14], built on a blockchain architecture, gives in [15] the value of establishing identity through the blockchain; of using a blockchain system to establish a verifiable public key for an identity. This is the basis of a PKI, and Buterin continues to outline processed for achieving PKI functionalities: revocation and key loss in particular.

Some earlier proposals were made, before the advent of blockchain, of decentralised PKI built through other means. KeyChains [16] is a peer-to-peer PKI system layered on top of a WoT trust model, and it is claimed in [16] that this is "the first PKI that is *truly decentralized* in all respects". To an extent, the PGP WoT model is decentralised - certificate *issuance* can be performed by any party and does not require any central authority. However, its storage and querying of certificates remains centralised, relying on central servers.

Also related is the use of blockchain in authenticated key exchange (AKE) protocols, for the establishment of secure communications channels without the involvement of a trusted third party. In [17] "Bitcoin-based AKE" is proposed - a "new category of AKE protocols that bootstrap trust entirely from the blockchain". This research is tailored in particular to the Bitcoin payment situation, in which there may be a requirement for secure communications between users post-transaction. The AKE proposal given does not require PKI.

There has been work in evaluating the security of blockchain in general; in [18] a formal study of distributed cryptography based on proof-of-work is given. There has also been some work in evaluating blockchain privacy; achieving privacy using the blockchain is somewhat counterintuitive, given that the blockchain is a public ledger and functions through public verification. Privacy is analysed for the Bitcoin blockchain in [19] and [20]. In the Bitcoin blockchain, nodes may remain pseudonymous, with no link to true identity; however, compromise of this pseudonymity is in some cases possible. In [21], it is shown that some Bitcoin transactions can be de-anonymised based on the flow of transactions: although account identities are anonymous, transactions are completely public, and any party can view the details of transactions between accounts (time and contents of transaction, amount of bitcoin, and sending and receiving Bitcoin addresses).

Some solutions for providing privacy in blockchain have been proposed. For the Bitcoin blockchain, CoinJoin[22] anonymises transactions through a coin mixing technique. Buterin, again, achieves a certain level of privacy for a blockchain-based secret-sharing decentralised autonomous organisation (DAO) by incorporating secure multiparty computation, to achieve"a blockchain-like system which offers decentralized control not just over the right to update the state, but even over the right to access the information at all" and details this in [23]. In [24], the authors develop the Enigma platform for decentralised computation and storage with guaranteed privacy. The network is controlled by an external blockchain, while computation is based on secure multiparty computation, and storage in a hash table for secret-shared data.

7.2 PKI Security and Privacy

Work has been done in improving PKI. The areas for improvement vary, and in the following we give work in improving various aspects of PKI - in particular, Certificate Transparency, prevention of man-in-the-middle attacks, and privacy. Problems with PKI have been highlighted: the risks associated with PKI are enumerated in [5]: particularly, the necessity to "trust" a CA and its practice and the protection of private keys.

Certificate Transparency (CT) was conceived by Google in 2011 as a method of monitoring digital certificates. Certificates are publicly logged and monitored, allowing the identification of rogue CAs and certificates by the domain owners themselves. CT has been implemented by DigiCert since 2013, and is running as a pilot by Google. In Section 4 we show that many of the problems with PKI that CT aims to eliminate can be addressed through the decentralisation of PKI.

We now give related work on the requirement for privacy in some PKI, the applications of such PKIs, and methods for achieving this "privacy-awareness".

In [9], Brands details the requirement for privacy in PKI. Privacy-aware PKIs are proposed in [25] and [26]. In [25], Ren et al. focus specifically on the problem of user revocation for privacy-aware PKI. The PKI is made privacy-aware through a group signature scheme. In [26], a privacy-aware PKI is proposed as a extension of an anonymous credentials system.

Anonymous public key solutions are explored in [27], which addresses the construction and applications of anonymous public key certificates. Untraceability is addressed in [28] for RFID, in which each peer computes its own anonymous public key and corresponding certificate; in this way the computation of public keys and certificates is somewhat *distributed* or *decentralised*. In [29], focus is on the protection of anonymity of *receivers* in networks, through the receiver's creation for itself of many separate anonymous and unlinkable "identities".

There is work in privacy-aware PKI for ubiquitous systems. In [30], a pseudonymous PKI is given for ubiquitous computing; in [31] privacy-aware ubiquitous systems are further detailed. [30] gives a pseudonymous alternative to conventional PKI, with focus on retaining the non-repudiation property provided by conventional PKI. In [32] a flexible, privacy-preserving authentication system for ubiquitous computing environments is proposed, and [33] explores challenges in providing security and privacy in pervasive computing environments.

For vehicular networks, there is also much work in privacy-awareness; in particular, the prevention

of tracking of vehicles through the tracking of their public keys. This is usually addressed through the frequent updating of unlinkable short-term public keys ([34]), and the employment of group signature schemes.

8 Conclusions and Further Work

Building PKI on the blockchain is a viable alternative to the conventional CA and WoT approaches, and offers desirable security properties. The blockchain operates in a decentralised way, and is a reliable record of transactions, making it a natural solution to many of the issues with conventional PKI: single points of failure and transparency - that all participating parties have the capability to check the transaction record - in particular are provided by the blockchain approach. Existing blockchain-based PKI proposals are built as transparent records of transactions, and do not provide the option to retain levels of privacy.

Privacy-awareness is required of PKI in present and emerging applications; in particular, mobile and vehicular networks and pervasive computing require PKIs in which entities may remain anonymous or pseudonymous network members. In our proposal for a blockchain PKI, we achieve total anonymity at some security cost: as explained in Section 6, network members may tamper in the short term with the online public keys of others. A slightly lower level of privacy can be achieved using attestation by a trusted group of "neighbours" to verify key changes against identity at each update, and security of the PKI is improved by this adjustment.

Our proposal and analysis show that privacy-aware PKI can be built on the blockchain, and desirable properties absent from conventional PKI - decentralised control of access to information; unlinkable key updates - can be given by this construction to fit the requirements of differing emerging applications that require some level of privacy-awareness.

As further work, we point to the further development of the framework to fit specific use cases; the improvement of the efficiency of the scheme, approaches to which are suggested but the detailing of which is outside the scope of this work.

References

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 1(2012):28, 2008.

[2] Conner Fromknecht, Dragos Velicanu, and Sophia Yakoubov. A decentralized public key infrastructure with identity retention. Technical report, Cryptology ePrint Archive, Report 2014/803, 2014. http://eprint. iacr. org, 2014.

[3] Bitcoin developer guide, https://bitcoin.org/, accessed on 10/09/2015 at 21:57.

[4] Namecoin, https://namecoin.info/, accessed on 17/09/2015 at 10:31.

[5] Carl Ellison and Bruce Schneier. Ten risks of pki: What you're not being told about public key infrastructure. *Comput Secur J*, 16(1):1–7, 2000.

[6] Google certificate transparency project, http://www.certificate-transparency.org/, accessed on 22/08/2015 at 14:06.

[7] Conner Fromknecht, Dragos Velicanu, and Sophia Yakoubov. Certcoin: A namecoin based decentralized authentication system 6.857 class project. 2014.

[8] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[9] Stefan A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, Cambridge, MA, USA, 2000.

[10] Denis Rystsov. Nidaba: a distributed scalable pki with a stable price for certificate operations.

[11] Sean Pearl. Distributed public key infrastructure via the blockchain. 2015.

[12] Greg Slepak. Dnschain + okturtles, https://okturtles.com/other/dnschain_okturtles_overview.pdf, accessed on 17/09/2015 at 10:31.

[13] okturtles foundation, https://okturtles.com/, accessed on 24/08/2015 at 10:57.

[14] Ethereum, https://www.ethereum.org/, accessed on 24/08/2015 at 10:20.

[15] Vitalik Buterin. Visions, part 1: The value of blockchain technology, https://blog.ethereum.org/2015/04/13/visions-part-1-the-value-of-blockchain-technology/, accessed on 24/08/2015 at 10.30.

[16] Ruggero Morselli, Bobby Bhattacharjee, Jonathan Katz, and Michael A Marsh. Keychains: A decentralized public-key infrastructure. 2006.

[17] Patrick McCorry, Siamak F Shahandashti, Dylan Clarke, and Feng Hao. Authenticated key exchange over bitcoin.

[18] Marcin Andrychowicz and Stefan Dziembowski. Distributed cryptography based on the proofs of work. Technical report, Cryptology ePrint Archive, Report 2014/796, 2014. http://eprint. iacr. org.

[19] Elli Androulaki, Ghassan O Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. Evaluating user privacy in bitcoin. In *Financial Cryptography and Data Security*, pages 34–51. Springer, 2013.

[20] Jaume Barcelo. User privacy in the public bitcoin blockchain.

[21] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M

Voelker, and Stefan Savage. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 127–140. ACM, 2013.

[22] Coinjoin, https://bitcointalk.org/index.php?topic=279249.0, accessed on 10/09/2015 at 21:57.

[23] Secret sharing daos: The other crypto 2.0, https://blog.ethereum.org/2014/12/26/secret-sharing-daos-crypto-2-0/, accessed on 19/08/2015 at 16:17.

[24] Guy Zyskind, Oz Nathan, and Alex Pentland. Enigma: Decentralized computation platform with guaranteed privacy. *arXiv preprint arXiv:1506.03471*, 2015.

[25] Wei Ren, Kui Ren, Wenjing Lou, and Yanchao Zhang. Efficient user revocation for privacy-aware pki. In *Proceedings of the 5th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, page 11. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[26] Pino Persiano and Ivan Visconti. An anonymous credential system and a privacy-aware pki. In *Information Security and Privacy*, pages 27–38. Springer, 2003.

[27] Kazuomi Oishi, Masahiro Mambo, and Eiji Okamoto. Anonymous public key certificates and their applications. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 81(1):56–64, 1998.

[28] Giuseppe Ateniese, Jan Camenisch, and Breno de Medeiros. Untraceable rfid tags via insubvertible encryption. In *Proceedings of the 12th ACM conference on Computer and communications security*, pages 92–101. ACM, 2005.

[29] Brent R Waters, Edward W Felten, and Amit Sahai. Receiver anonymity via incomparable public keys. In *Proceedings of the 10th ACM conference on Computer and communications security*, pages 112–121. ACM, 2003.

[30] Ke Zeng. Pseudonymous pki for ubiquitous computing. In *Public Key Infrastructure*, pages 207–222. Springer, 2006.

[31] Marc Langheinrich. Privacy by design—principles of privacy-aware ubiquitous systems. In *Ubicomp 2001: Ubiquitous Computing*, pages 273–291. Springer, 2001.

[32] Jalal Al-Muhtadi, Anand Ranganathan, Roy Campbell, and M Dennis Mickunas. A flexible, privacy-preserving authentication framework for ubiquitous computing environments. In *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*, pages 771–776. IEEE, 2002.

[33] Roy Campbell, Jalal Al-Muhtadi, Prasad Naldurg, Geetanjali Sampemane, and M Dennis Mickunas. Towards security and privacy for pervasive computing. In *Software Security—Theories and Systems*, pages 1–15. Springer, 2003.

[34] Ahren Studer, Elaine Shi, Fan Bai, and Adrian Perrig. Tacking together efficient authentication, revocation, and privacy in vanets. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON'09. 6th Annual IEEE Communications Society Conference on*, pages 1–9. IEEE, 2009.