Beyond the Hype: On Using Blockchains in Trust Management for Authentication*

Nikolaos Alexopoulos, Jörg Daubert, Max Mühlhäuser, and Sheikh Mahbub Habib

Telecooperation Lab
Technische Universität Darmstadt
Hochschulstraße 10, Darmstadt D-64289, Germany

{alexopoulos, daubert, max, sheikh}@tk.tu-darmstadt.de

Abstract

Trust Management (TM) systems for authentication are vital to the security of online interactions, which are ubiquitous in our everyday lives. Various systems, like the Web PKI (X.509) and PGP's Web of Trust are used to manage trust in this setting. In recent years, blockchain technology has been introduced as a panacea to our security problems, including that of authentication, without sufficient reasoning, as to its merits.

In this work, we investigate the merits of using open distributed ledgers (ODLs), such as the one implemented by blockchain technology, for securing TM systems for authentication. We formally model such systems, and explore how blockchain can help mitigate attacks against them. After formal argumentation, we conclude that in the context of Trust Management for authentication, blockchain technology, and ODLs in general, can offer considerable advantages compared to previous approaches. Our analysis is, to the best of our knowledge, the first to formally model and argue about the security of TM systems for authentication, based on blockchain technology. To achieve this result, we first provide an abstract model for TM systems for authentication. Then, we show how this model can be conceptually encoded in a blockchain, by expressing it as a series of state transitions. As a next step, we examine five prevalent attacks on TM systems, and provide evidence that blockchain-based solutions can be beneficial to the security of such systems, by mitigating, or completely negating such attacks.

1 Introduction

We live in an increasingly interconnected world, where online interactions are a common occurrence in the daily routine of most individuals. Billions of interactions, possibly of a sensitive nature, take place every day around the world. The security of these interactions is essential for the orderly operation of organizations, as well as for the well-being of people.

Authentication is a major enabler of security on the Internet. In this context, authentication corresponds to a mechanism that verifies the identities of interacting entities. It is a requirement of almost all secure online interactions, ranging from email exchange to banking transactions. As an example, authentication mechanisms as part of the TLS/SSL protocol [DR08] are used each time a user browses a website through https [Res00]. Through the use of public key cryptography, authentication of an entity

*A version of this paper will appear at IEEE TrustCom-17

equates to verifying the correctness of the binding between a public key and an identity, which in our context also includes related attributes of the entity (see section 2). For the remainder of the paper this will be called a *public-key-to-id* binding.

Trust is a vital component of authentication systems. The electronic nature of online interactions means, that physical identification using, e.g. state issued id cards and passports, is not possible, or even appropriate. Furthermore, in many instances, identification includes more than just name verification, e.g. in an online marketplace scenario, the client would need to know if the interacting party is actually the shop with the corresponding physical address that it claims, possibly along with other relevant attributes, such as certifications of the shop's quality. Therefore, entities that have not had direct physical experience with the holders of the public keys that they want to identify, need to rely on others to authenticate the public-key-to-id bindings[1]. Entities trust that other entities are *proficient* in the identification process and *honest* in the authentication process of third parties. This trust is then disseminated in the network through chains of trust expressed via cryptographic credentials, thus making possible the authentication of entities without physical interaction. Two of the most widely used systems that achieve this goal are the web X.509 PKI ([CFS+03, CSF+]) and PGP's Web of Trust [Zim95]. The former follows the centralized hierarchical approach, with designated root certification authorities (CAs) trusted a priori by browsers and operating systems. These can in turn delegate this trust to other CAs and so on, to form chains. The latter is based on a decentralized approach and allows users to authenticate other users by forming trust chains through their social relations in the real world.

Real-world issues [Bri11, Art11, Goo16] have highlighted the need for more secure and transparent authentication systems. Recently, and in accordance with the need for decentralized and secure authentication, preliminary solutions using blockchain technology have been proposed to solve some aspects of the problem ([nam, FVY14, Sle13, WA15, ANSF16, TD16]) with a varying degree of success. For example, Namecoin [nam] offers a decentralized namespace supported by a dedicated cryptocurrency. In [WA15], the authors propose storing PGP keys and signatures on the bitcoin blockchain, while [ANSF16] introduces a distributed naming and storage service that can be implemented on top of any suitable blockchain. Blockchains and Open Distributed Ledgers (ODLs) in general, as explained in Section 2, can potentially offer security and transparency by design, to the problem, and therefore enhance the security of authentication infrastructures. However, the solutions mentioned above, offer ad hoc advantages compared to the traditional schemes, that are in general not formally studied.

In this paper, we investigate whether or not, the hype [Dic17] surrounding blockchain technology, as a solution to security problems, has concrete backing. We limit our investigation to Trust Management (TM) systems for authentication and we ask the following question: *Are there merits to using blockchain technology to enhance the security of TM systems for authentication? - and if so: which ones?*

To answer this question, we develop an abstract model for TM systems for authentication, based on graph theory. Our model's advantage in comparison with previous work lies with its generic nature and simplicity. It enables us to argue about TM systems in a uniform way and express preperties not specific to any one system. We also provide a way to interpret the use of blockchain technology in this context, by encoding our model to a state machine, corresponding to a secure blockchain. Then, we present a set of five prevalent attacks against authentication systems, and show that blockchain technology can in a great extent enhance the security of these systems against those attacks. Therefore, as a result of formal reasoning, we conclude that the answer to the question posed above, is an affirmative one. Our work is novel and significant in that regard, as this area has not been investigated in the past in a generic manner, and is an important step towards understanding the importance of using this pervasive technology to its full potential.

[1] We refer to systems that rely on trust for authenticating third parties as: *"Trust Management systems for authentication"*

1.1 Our Contributions

The main question we address with this paper is, whether or not blockchain technology can enhance the security of Trust Management systems for authentication. On the way to affirmatively answering the question above, we make a series of contributions. Namely we present:

- A graph theoretic model of trust networks for authentication and how to encode it in the blockchain.
- An analysis of selected attacks against trust networks and corresponging defenses enabled by blockchain technology.
- An insight into future directions for secure and decentralized TM.

1.2 Paper Organization

We begin by going through notation and required preliminary knowledge in Section 2. Then, we present our model in Section 3, followed by an analysis of attacks and defenses in Section 4. Finally, we discuss our vision for future work in this area in Section 5 and conclude in Section 6.

2 Notation and preliminaries

Before we introduce our model, we present the notation used throughout the paper, as well as some important background information.

Notation A *TM system* for authentication is a system managing digital representations of social trust to enable decisions regarding public-key-to-id bindings. An *entity* is a public key participating in the system. A *chain* is a path of directed edges of a (multi)graph. Moreover, $a := b$ denotes that a is defined as b. The adversary is denoted with $\mathcal{A}$.

Public key cryptography Public key cryptography is the area of cryptography dealing with asymmetric cryptographic systems. It is based on the notion that each entity has a pair of cryptographic keys, a public one and a secret one. The public key of the entity can be used to encrypt messages that only the possessor of the corresponding secret key can decrypt. Furthermore, the possessor of a secret key can authenticate messages by digitally signing them with it, and all other entities can verify the signature with knowledge of the public key.

Identity The Oxford dictionary defines identity (id) as "The characteristics determining who or what a person or thing is"[2]. In the digital world, the identity associated with a public key can be perceived as a set of attributes the holder of the corresponding private key has [ISO11]. In other words, who or what the holder of the secret key is. For example, in the case of browsing the web, identity corresponds to the domain name and the organization bound to a public key, whereas in the case of email exchange, identity corresponds to the email address and person name bound to a public key. An identity can be associated with multiple public keys, as a person may want to use different keys for different purposes.

Blockchain Blockchain technology, as introduced with Bitcoin in [Nak08], offers an open, secure and distributed transaction ledger (ODL). As realizations of the technology focus on implementing currency systems and are based on cryptographic primitives, they are known as *cryptocurrencies*. The basic idea behind the design is facilitating decentralized consensus, that is, making it possible for a network of unknown participants to jointly decide on a global view and ordering of transactions. Transactions are grouped in blocks and in each round, a participant is elected to propose a valid block. The election mechanism is in most cases based on solving a computational puzzle (proof-of-work), but other approaches like proof-of-stake [But13] have been proposed. Each participant then inspects the proposed block and if it is valid, she includes it in her blockchain. We refer readers not familiar with blockchain technology

[2] https://en.oxforddictionaries.com/definition/identity

to [BMC+15] for an overview of bitcoin and cryptocurrencies. Security of the blockchain is guaranteed under well defined assumptions, relating to the ratio of honest and malicious users, as well as to the puzzle's hardness. Specifically, security in Bitcoin is guaranteed assuming an honest participating majority, in addition with a puzzle hard enough so that the average time needed to solve it is much less than the time needed for messages to propagate in the network. The interested reader is referred to [GKL15] for further reading. In this paper, we view the blockchain as a probabilistic state machine that converges after a number of time steps, similar to how it is described in [Woo14] and [SY16]. All entities view the blockchain as an append-only history of state transitions.

3 A model of a TM system for authentication

A general approach is necessary to address the common security issues of authentication TM systems. Previous attempts to model TM systems in other contexts use either graph theory [JGK06] or mathematical logic [HRMV13, MP07]. In this paper, we choose to use graph theory to model and reason about the security of such systems, as this representation is intuitive and can support arguments concerning the network connections of the nodes. Our representation is compatible with the seminal work of Maurer [Mau96] and its extension [MS05]. We begin by defining the basic concepts of our model, then we consider the capabilities of possible adversaries, and finally we give examples by instantiating our model with OpenPGP and X.509 to show its generality.

3.1 The Model

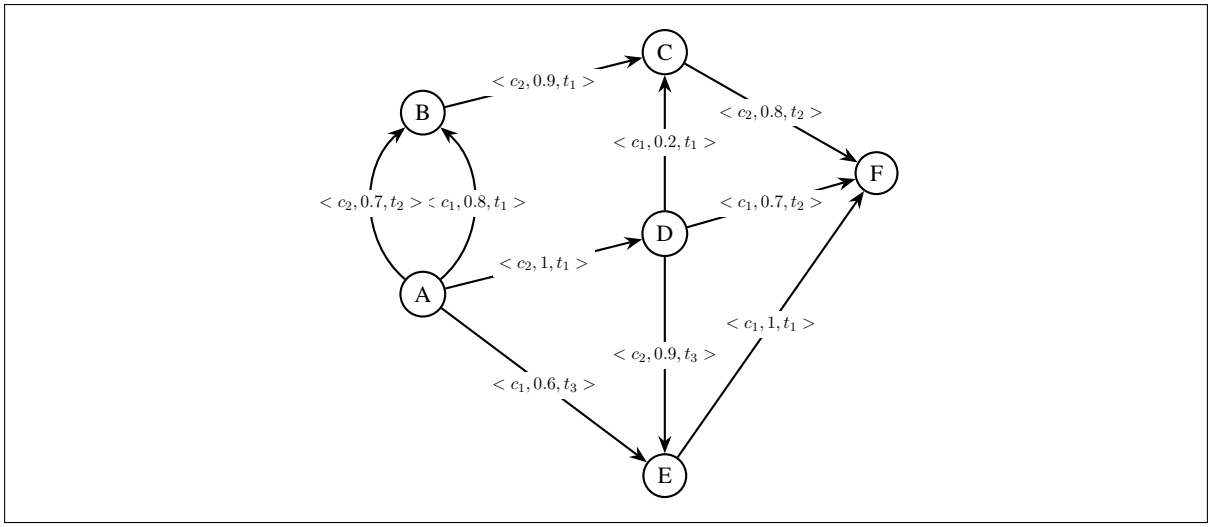
We model a TM system for authentication, as a directed graph with vertices representing entities that participate in the system, and edges representing trust relations. Entities are perceived as public keys participating in the system. A trust relation is the core basic concept of any TM system and expresses the trust a trustor puts into a trustee.

Definition 1 (Trust relation)**.** *A Trust relation (TR) is a sequence (tuple) $\langle A, B, c, v, \alpha, t\rangle$ where:*

- *A is the trustor, i.e. the entity that expresses her trust towards another entity.*
- *B is the trustee, i.e. the entity that is the subject of the trust expressed by A.*
- *c is the context of the relation, i.e. an identifier of the class the relation belongs to, along with relevant information.*
- *$v \in [0, 1]$ is the actual trust value (assigned to B by A. In some cryptographic systems, like the X.509 PKI, it is implicitly assumed to be 1 or 0, i.e. total or no trust. Alternatively, it can take multiple discrete values, e.g. see Fig. 2 for OpenPGP. In general, it can take arbitrary continuous or discrete values, which in our model are normalized in the interval $[0, 1]$.*
- *α is a set of cryptographic artifacts, most commonly digital signatures, that are used to assure integrity and authenticity of the relation.*
- *t is a logical time component, marking the time the relation was last updated. We assume a partial ordering of events in the distributed system [Lam78], i.e. events at time $t+i$, $i > 0$ are considered to have happened after events at time t.*

Regarding the context c of the trust relation, for the purpose of authentication two context classes are generally relevant. First, the context class of *validity*. In this context class, c includes an identity, i.e. a set of attributes, and the trust relation expresses the belief that A has about the binding of the identity

Figure 1: An example TM network. We omit the two first members of the tuple, as the information is conveyed by the graphical representation. Also, α is omitted for brevity. The context c, is depicted with its class as subscript. We can see a variety of trust relations of different contexts and timestamps, forming a (multi)graph.



in c, with B. The second relevant context class, the *authenticator trust* context class, expresses the trustworthiness of B as an authenticator of unknown entities. It is possible that A trusts B in the context of validity, i.e. she knows the identity of B, however she does not trust B to authenticate other entities. Hence, differentiating among different trust contexts is necessary. A trust relation can also express negative trust information, like the revocation of a key, when e.g. the secret key is lost or the identity of an entity changes. Revocation information is represented by setting $v = 0$. Complete uncertainty can be modelled by setting $v = 0.5$.

To express the whole ecosystem of trust relations, we define the TM network, which includes all trust relations between participating entities.

Definition 2 (TM network)**.** *A TM network or trust graph is a directed multigraph $G = (V, E)$ where:*

- *Each $v \in V$ is an entity, e.g. certification authority, physical person etc.*
- *Each $e \in E$ is labeled with a trust relation.*

An example of a TM network is given in Fig. 1. Finally, the goal of a TM network is to enable the assessment of trust in different contexts, by taking into account the propagation of trust along edges.

Definition 3 (Trust assessment)**.** *A trust assessment $T^c_{A\to B}$ is the result of the calculation of trust A puts into B in a given context c, defined as:*

$$T^c_{A\to B} := \mathcal{P}(c, H), \quad \mathcal{P} : c \times H \subseteq G \to [0, 1]$$

where $\mathcal{P}$ is a program that takes as input a trust network in the form of a network subgraph H, which represents the view of the graph by A, and outputs a trust value in a given context. This trust value can again take binary, discrete or continuous values and is normalized in the interval $[0, 1]$. The subgraph H is the trust view of the trust assessment and includes the vertices and edges used for the computation by the program $\mathcal{P}$. The program $\mathcal{P}$ expresses the way trust propagates in the network, e.g. via trust chains.

3.2 The Adversary

In this paper, we consider an adversary $\mathcal{A}$ that attacks the TM system in order to impersonate an honest entity, i.e. claim an identity that is not rightfully his. In computer security terminology, this is a Man in the Middle (MITM) attack. Depending on the attack, the adversary might also want to remain undetected. We assume a powerful adversary that can arbitrarily add and remove labeled edges and vertices from the network graph G. In literature, this adversary is known as an *arbitrary adversary* [MT16] and is equivalent to an adversary that controls a subset of the entities of the system, along with a subset of the communication channels. Finally, we assume that there exist cryptographic primitives, namely digital signatures and encryption schemes, that are secure against $\mathcal{A}$, i.e. the adversary is computationally bounded as per standard security literature.

3.3 Trust assessment security

As TM systems are generally used to assist in decision making, their security could be defined as, whether or not the decisions made using them are the correct ones. For example, if access to a resource is only granted to an authorized party, or a given public key actually belongs to a physical entity. The latter definition is the one we adopt in this paper. However, these decisions depend not only on the TM network infrastructure, but also on other factors, like the program $\mathcal{P}$ used to derive the trust assessment, as well as the decision strategy of the user. Therefore, the security of a TM system is conceptually decoupled from the decision made, and resolves around computing a correct trust assessment, even when some entities and communication links are controlled by an adversary $\mathcal{A}$.

3.4 Examples

We showcase the generality of our model by fitting it to two representative systems following different approaches.

3.4.1 OpenPGP Web of Trust

The Open Pretty Good Privacy protocol (OpenPGP) is an open source version of the initial PGP protocol and the de facto standard for decentralized exchange of encrypted messages. It employs an anarchic Web of Trust (WoT) to (potentially mutually) authenticate parties. Trust information is stored in specified keyservers that are tasked with making it available upon request. More details about the state and the security of the OpenPGP WoT can be found in [UHHC11] and [BDFPS15]. The realization of our model for the OpenPGP WoT can be seen in Fig. 2.

3.4.2 X.509 PKI

The X.509 standard PKI is a representative of the centralized model for authentication and embraces a hierarchical trust structure. It is the standard method for authenticating web domains, and therefore the most widely used authentication system today. Several "root" certification authorities are pre-trusted by default by browser vendors, e.g. Mozilla products ship with $\approx$ 170 root CAs [moz]. These root CAs can in turn delegate this trust to any other CA, and any CA can authenticate any given identity (usually domain name). The realization of our model for the X.509 PKI can be seen in Fig. 3

Remark 1. The goal of this section is not to exhaustively specify existing authentication systems in our model, but to showcase the generality of our abstraction.

Figure 2: OpenPGP Web of Trust instantiation

i. Trust relations are signed certifications where:

- A is the trustor, i.e. the entity that expresses trust
- B is the trustee, i.e. the subject, toward which trust is expressed.
- $c \in \{\text{validity, trust}\}$. The system employs two contexts, one for expressing trust in the validity of a public-key-to-id binding and another expressing the level of trust in an entity as an authenticator.
- v can take different values according to the context. For the validity context, it takes values $v \in \{\text{full, marginal, untrusted, unknown}\}$ and for the trust context it takes values $v \in \{\text{ultimate, full, marginal, untrusted,undefined}\}$.
- α is a digital signature verifying A is the owner of the relation.
- t is the system time of the last update of the relation

ii. The trust assessment concerns the validity of a given *public-key-to-id* binding. More specifically, program $\mathcal{P}$ outputs the level of trust put on the validity of a binding, by forming certification chains. By default, PGP requires one fully (or ultimately) trusted signature or two marginally trusted ones to establish a key as valid. More details on the system are available in [CDF+07].

3.5 Modelling the use of blockchain

After describing our graph model for authentication TM systems, we proceed to provide a blockchain encoding of it. As mentioned in Section 2, we model the blockchain as a tamper proof, distributed state machine. Trust transactions either add, remove or modify an edge of graph G. Vertex existence is implied by the existence of edges. Each block committed to the blockchain represents a snapshot of the graph at a given time t, as it can be seen in Fig. 4. The snapshot is considered stable, i.e. tamper proof, after an adequate amount of time steps, e.g. in Bitcoin a transaction is assumed irreversible after 6 confirmations as a rule of thumb [bit]. In our reasoning about the use of blockchain, we assume entities accept only tamper proof graph snapshots, i.e. they take into account state transitions that are adequately deep in the chain, and therefore can be reversed with negligible probability.

To sum up, in this section, we introduced our working model of TM systems for authentication and how it can be conceptually realized by a blockchain. This lays the ground for the attacks and defences that follow in Section 4.

4 Attacks and defenses

Attacks described in this section, and generally attacks against authentication systems, are characterized as *impersonation attacks*. The goal of the adversary is to impersonate another entity, usually performed via a MITM attack. In this section, we present common and realistic attacks against authentication systems, and how blockchain technology enables defending against them. For an insight into other methods to mitigate these attacks we refer the readers to remark 2. We employ example graphs to better illustrate the attacks. The graphs follows the notation of Section 3, with the addition of dashed

Figure 3: X.509 PKI instantiation

i. Trust relations are signed certifications where:

- A is the trustor, i.e. a user that trusts a CA, or a CA.
- B is the trustee, i.e. the subject toward which trust is expressed.
- $c \in \{\text{certify, authenticate}\}$. The system employs two contexts. One for expressing the delegation of trust from a CA of a higher tier to one of a lower tier (certify) and one for expressing the trust a CA puts on a public-key-to-id binding of an entity-domain (authenticate).
- v is set to 1, i.e. total trust, for all trust relations expect revocation information, in which case
  $v = 0$.
- α is a digital signature verifying A is the owner of the relation.
- t is the system time of the last update of the relation

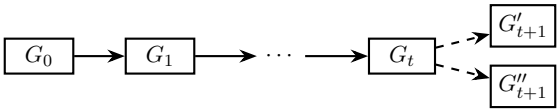
ii. The trust assessment concerns the validity of a given *public-key-to-id* binding. Program $\mathcal{P}$ works as follows: if there exists any chain of trust from an entity to another entity, where all but the last relation are of a certification context, i.e. $c =$ certify, and the last relation of the chain is of authentication context, i.e. $c =$ authenticate, then the trust put into the public-key-to-id binding is 1.

lines which represent trust relations that are part of a chain affected by the attack. Context notation is suppressed for simplicity. Colored vertices represent entities controlled by $\mathcal{A}$.

4.1 Stealthy targeted attack

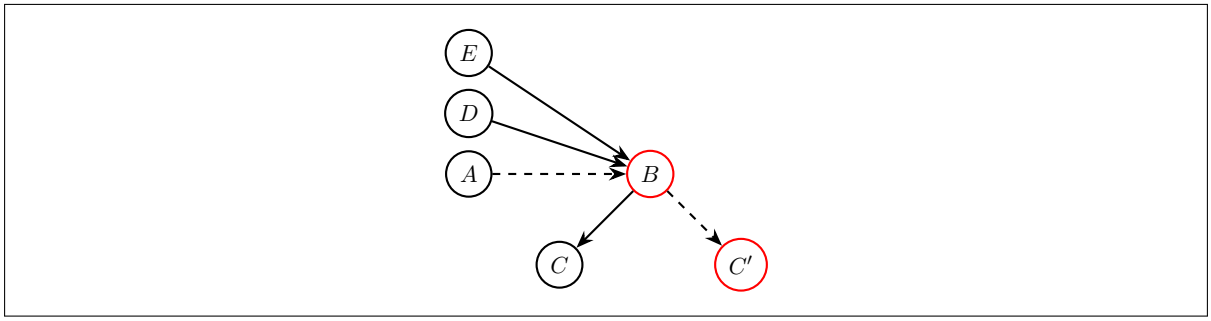
The attack: In this scenario, an adversarial entity forwards different trust relation information about a given subject, to the attack target, compared to the rest of the network. This way, $\mathcal{A}$ manipulates the target's trust assessment. An attack of this kind aims at mounting a MITM against a specific user while avoiding detection. The adversary controls a sufficient number of entities, that are directly or indirectly trusted by the target. An example of this class of attacks would be a malicious CA in a X.509 system, that wants to gain access to private information of a specific user. This attack is similar in nature to

Figure 4: Example of graph transitions on the blockchain. The initial state is the empty graph $G_0 = (\emptyset, \emptyset)$. The state machine has converged up to a point t in the future. A fork starts at block $t + 1$.



what is known as a *discrimination* [JG09, WMLZ14] or *conflicting behavior* [SHYL06] against trust evaluation systems. By providing malicious trust information only to a specific target, the adversary mounts a MITM attack against the target, that is difficult to detect, hence stealthy. An example can be seen in Fig. 5, where $\mathcal{A}$ controls B and mounts a MITM against A, by maliciously authenticating C' as having the identity that is associated with C. The rest of the TM network, i.e. D and E, continue to receive correct information from B and therefore cannot detect the attack.

Figure 5: Stealthy targeted attack example



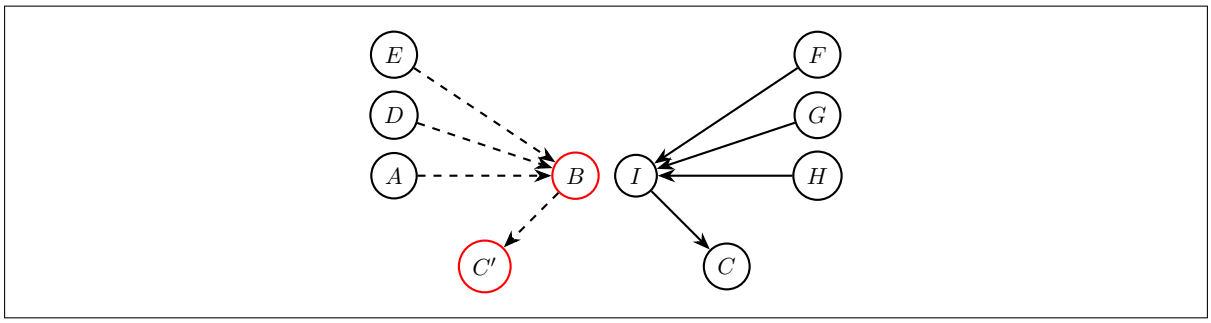
The defense: The success of attacks of this kind is based on the inability of a large portion of the network to detect the attack. Under the assumptions of our blockchain model for trust management 3.5, this attack is no longer possible. This is the case because having conflicting views of the trust relations at a given time t would mean having conflicting views of the trust graph G, which in turn would mean that a blockchain fork is present starting at a time $t' \leq t$. According to our assumptions about the security of the blockchain, this event is improbable, as entities are assumed to take into account blocks adequately deep in the chain. Therefore, the attack cannot take place with non negligible probability.

4.2 Double registration attack

The attack: In this scenario, the adversary wants to mount an attack against a set of targets by stealing the identity of an entity already participating in the system. An example would be, an entity with considerable capabilities, that succeeds in convincing a set of honest entities to authenticate the former as having the identity of a different user, already participating in the system. This impersonation attack could be achieved by issuing fake government credentials or tricking the authenticators into believing e.g. that a person represents an organization when this is not the case. An example of the attack can be seen in the TM network of Fig. 6, where entity B authenticates C' as having an identical identity to C. The partition of the trust network, makes the attack undetectable in the long run and entities A, D, E are affected, leading them to make wrong trust assessments. We note that binding two public keys with the same identity is not an attack by itself. To the contrary, it is a well established practice for individuals to have more than one public keys for different purposes, e.g. one key for signature verification and another one for encryption. However, the fact that knowledge of the different keys is not shared with the entire network is what makes this scenario an attack.

The defense: The success of this attack is based on the limited scope of view of the participating entities. Specifically, there exist two trust relations binding different public keys to the same identity and no honest entity has knowledge of both. Under the assumptions of our blockchain model, all participants have global view of the trust relations, i.e. they know the whole trust graph. Thus, two trust relations with the properties described above will be detected by every participant of the system and therefore

Figure 6: Double registration attack example

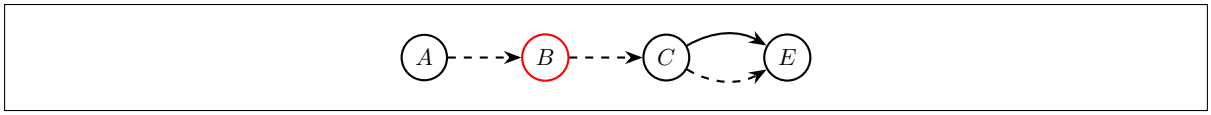


this attack is not possible. Detecting the attack naturally negates it, as the target should not consider the information provided by the attacker as valid.

4.3 Stale information attack

The attack: In this scenario, an entity makes a trust assessment based on information that is not up to date. The most common occurrence of this attack is when an entity makes a decision regarding the authentication of another entity, without taking into account relevant revocation information. In the example of Fig. 7, B forwards to A a trust relation of C that is not up-to-date. Specifically, there exist trust relations $TR_1 = \langle B, C, c_1, v_1, \alpha_1, t_1 \rangle$ and $TR_2 = \langle B, C, c_2, v_2, \alpha_2, t_2 \rangle$ with $t_1 < t_2$, so that A is aware of TR_1 but not TR_2. In the case that TR_2 expresses negative information about the validity of the key binding of E, then this attack can lead to a MITM against A. Imagine that C is a CA that revokes its authentication of E due to, e.g. the theft of E's secret key. Then, the adversary, by withholding this information from A leaves him vulnerable to an attack by the thief.

Figure 7: Stale information attack example



The defense: The success of this attack is based on the inability of an entity to get the latest updates on trust relations in the network, mainly due to network attackers. Our blockchain abstraction provides a global view of the trust graph and a partial ordering of trust relations with regard to time. Therefore, all entities with access to the blockchain, i.e. to any single honest entity in the network, are able to detect stale information and discard it accordingly, thus making the stale information attack impossible.

4.4 Denial of Service attack

The attack: The availability, i.e. usefulness, of a system is a major component of its ability to operate securely. Denial of Service (DoS) attacks may not affect the core security properties of a TMS, in the sense that no illegal operation will be carried out, however, the operation of the real-world applications depending on the TMS is disrupted in various degrees depending on the application. There is recent evidence of denial of service attacks on the internet's DNS infrastructure paralyzing a large portion of the system [dyn]. The authentication infrastructure is also a lucrative target for attackers. Certificate

revocation services (X.509), as well as keyservers (PGP) could be targeted in an attempt to hinder the functionality of the system.

The defense: The decentralized nature of blockchain technology can make DoS attacks against specific entities ineffective. Specifically, under the assumptions of our ideal blockchain model, any entity with network access to the blockchain overlay is able to retrieve the latest, converged state of the system and thus make informed trust assessments. The peers providing the information need not be trusted, as security is guaranteed under the assumptions outlined in Section 2. Network attacks against the blockchain itself have been studied in [VTM14] and [HKZG15] for the case of Bitcoin. These attacks generally require much stronger adversaries than the ones considered in this paper, and are also accompanied with mitigation measures.

4.5 Censorship attack

The attack: A censorship attack is essentially a DoS variant. However, as the objective and the motives differ from the general case of a DoS, we handle this attack individually. In this scenario, a powerful adversary forbids other entities within his legal jurisdiction to provide authentication services to specific entities. Centralized systems (e.g. X.509) are inherently susceptible to this attack, however decentralized systems that rely on keyservers for trust information availability (e.g. PGP) can also be affected.

The defense: Blockchain-based systems are assumed to be inherently censorship resistant, meaning that no single entity, or small group thereof, can prevent other entities from submitting transactions to the ledger. An adversary would need to control the majority of participating nodes in order to mount this attack, which is contrary to our security assumptions. However, when considering rational adversaries, an entity may not want to risk producing a block that is in turn rejected by the network, and therefore this attack may be more likely in this setting. However, this adversary model is out of the scope of this paper.

Remark 2. Different methods to enhance the security of the web PKI (X.509) system have been proposed in recent years [GSM+13, WAP08, LLK13, MBB+15, STV+16, BVC+14, CBV+15]. Most of these approaches distribute trust among monitors and witnesses to keep the CAs honest. These enhancements are either inherently offered by blockchain constructions by design or can be implemented upon them.

4.6 Section summary

In this section, we presented five prevalent attacks against TM systems for authentication. Our model (Section 3), allowed us to reason about such systems in a generic manner, and in combination with our blockchain model, enabled us to conceptually analyze the use of blockchain technology for Trust Management. We showed that blockchain technology can mitigate, or completely negate the attacks described in this section, therefore providing evidence of its merit in this area.

5 Discussion and challenges

In this paper, we showcased the advantages of using ODLs in TM for authentication. Despite ongoing research making first steps towards this direction, there are still a multitude of problems to be resolved before this technology can be widely deployed. In our blockchain abstraction (Section 3.5), we implied that the whole set of state-changing transactions is stored on the blockchain. The size of this construct will quickly cause the blockchain to bloat [Wag14], thus continuously increasing the capacity needs of participating entities. Moreover, as the capacity and processing needs are going to be substantial, thin clients [FMR+16] will need to be deployed on resource-constrained devices. When considering the

security of the blockchain itself, ardent theoretical and experimental evaluation should accompany any proposed design. Regarding the last point, the incentive mechanism used for blockchain participation will greatly influence the security of the blockchain and the problem of designing it is of great significance. Privacy is another major concern that has to be resolved, as balancing the need for transparency and user privacy is a universal problem. Specifically, the transparency that comes hand in hand with the use of blockchain technology can provide information about the social relations and interactions of parties, that needs to be protected. Another key design consideration is that of the type of blockchain to be used. Public/open/permissionless designs, like the ones adopted by Bitcoin and Ethereum, enable the open participation of all entities that want to contribute to the system. In this case, it will be important to hold the participants accountable for their actions, so that they face repercussions if they misbehave. On the other hand, so called permissioned/consortium blockchain designs, as the one used in [fab], can offer some advantages regarding accountability, however they lack the decentralized nature of permissionless systems.

6 Conclusion

In this paper, we showed that ODLs, as implemented by blockchain technology, can *by design* enhance the security of authentication infrastructures. We introduced an abstract, graph-theoretic model of TM systems for authentication, and a matching blockchain model. Our blockchain model for Trust Management is a distributed, probabilistic state machine. States of the machine correspond to snapshots of our graph-theoretic model. We then highlighted five prevalent attacks and showed that under the assumptions of our blockchain abstraction, they can be alleviated by encoding the trust information in a secure blockchain. Specifically, the fact that blockchain enables all participants to have a consistent, transparent view of the trust network, solves many of the issues of traditional authentication systems. By doing this, we showcased that *blockchain* is not just a buzzword in the case of Trust Management systems. There is genuine merit in applying blockchain-based designs in this field of network security, and that can lead to more secure and trustworthy systems overall. The authentication problem is expected to become even more challenging, as new paradigms, like the IoT, come into reality and systematically exploring the advantages of blockchain technology can provide new solutions. While blockchains and ODLs in general, can offer concrete advantages compared with traditional approaches to the problem of authentication, there are still important issues to be resolved in order for this technology to yield real-world results.

Acknowledgments

This work has been co-funded by the DFG as part of project S1 within the CRC 1119 CROSSING and as part of project D.4 within the RTG 2050 “Privacy and Trust for Mobile Users”.

References

[ANSF16] Muneeb Ali, Jude Nelson, Ryan Shea, and Michael J Freedman. Blockstack: A global naming and storage system secured by blockchains. In *2016 USENIX Annual Technical Conference (USENIX ATC 16)*, pages 181–194. USENIX Association, 2016.

[Art11] C Arthur. Diginotar ssl certificate hack amounts to cyberwar, says expert. *The Guardian. http://www. theguardian. com/technology/2011/sep/05/diginotar-certificate-hack-cyberwar*, 2011.

[BDFPS15] Alessandro Barenghi, Alessandro Di Federico, Gerardo Pelosi, and Stefano Sanfilippo. Challenging the trustworthiness of pgp: Is the web-of-trust tear-proof? In *European Symposium on Research in Computer Security*, pages 429–446. Springer, 2015.

[bit] Bitcoin wiki.

[BMC+15] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A Kroll, and Edward W Felten. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 104–121. IEEE, 2015.

[Bri11] Peter Bright. How the comodo certificate fraud calls ca trust into question. *arstechnica*, 2011.

[But13] Vitalik Buterin. What proof of stake is and why it matters. *Bitcoin Magazine, August*, 26, 2013.

[BVC+14] Johannes Braun, Florian Volk, Jiska Classen, Johannes Buchmann, and Max Mühlhäuser. Ca trust management for the web pki. *Journal of Computer Security*, 22(6):913–959, 2014.

[CBV+15] Jiska Classen, Johannes Braun, Florian Volk, Matthias Hollick, Johannes Buchmann, and Max Mühlhäuser. A distributed reputation system for certification authority trust management. In *Trustcom/BigDataSE/ISPA, 2015 IEEE*, volume 1, pages 1349–1356. IEEE, 2015.

[CDF+07] J Callas, L Donnerhacke, H Finney, D Shaw, and R Thayer. Rfc 4880-openpgp message format. *Informe técnico, Internet Engineering Task Force (IETF)*, 2007.

[CFS+03] Santosh Chokhani, Warwick Ford, Randy Sabett, Charles Merrill, and Stephen Wu. Internet x. 509 public key infrastructure certificate policy and certification practices framework. Technical report, 2003.

[CSF+] D Cooper, S Santesson, S Farrell, S Boeyen, R Housley, and W Polk. Rfc 5280: Internet x. 509 public key infrastructure certificate and certificate revocation list (crl) profile. internet engineering task force (ietf), 2008.

[Dic17] Ben Dickson. Blockchains brilliant approach to cybersecurity. *Venturebeat. http://venturebeat.com/2017/01/22/blockchains-brilliant-approach-to-cybersecurity/*, 2017.

[DR08] Tim Dierks and Eric Rescorla. Rfc 5246: The transport layer security (tls) protocol. *The Internet Engineering Task Force*, 2008.

[dyn] 2016 dyn cyberattack.

[fab] Hyperledger fabric.

[FMR+16] Davide Frey, Marc X Makkes, Pierre-Louis Roman, François Taïani, and Spyros Voulgaris. Bringing secure bitcoin transactions to your smartphone. In *Proceedings of the 15th International Workshop on Adaptive and Reflective Middleware*, page 3. ACM, 2016.

[FVY14] Conner Fromknecht, Dragos Velicanu, and Sophia Yakoubov. A decentralized public key infrastructure with identity retention. *IACR Cryptology ePrint Archive*, 2014:803, 2014.

[GKL15] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 281–310. Springer, 2015.

[Goo16] Dan Goodin. Firefox ready to block certificate authority that threatened web security. *Ars Technica. https://arstechnica.com/security/2016/09/firefox-ready-to-block-certificate-authority-that-threatened-web-security/*, 2016.

[GSM+13] Slava Galperin, Stefan Santesson, Michael Myers, Ambarish Malpani, and Carlisle Adams. X. 509 internet public key infrastructure online certificate status protocol-ocsp. 2013.

[HKZG15] Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. Eclipse attacks on bitcoin's peer-to-peer network. In *USENIX Security*, pages 129–144, 2015.

[HRMV13] Sheikh Mahbub Habib, Sebastian Ries, Max Mühlhäuser, and Prabhu Varikkattu. Towards a trust management system for cloud computing marketplaces: using caiq as a trust information source. *Security and Communication Networks*, pages 1–16, 2013.

[ISO11] Information technology – Security techniques – A framework for identity management – Part 1: Terminology and concepts. Standard, International Organization for Standardization, December 2011.

[JG09] Audun Jøsang and Jennifer Golbeck. Challenges for robust trust and reputation systems. In *Proceedings of the 5th International Workshop on Security and Trust Management (SMT 2009), Saint Malo, France*, 2009.

[JGK06] Audun Jøsang, Elizabeth Gray, and Michael Kinateder. Simplification and analysis of transitive trust networks. *Web Intelligence and Agent Systems: An International Journal*, 4(2):139–161, 2006.

[Lam78] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978.

[LLK13] Ben Laurie, Adam Langley, and Emilia Kasper. Certificate transparency. Technical report, 2013.

[Mau96] Ueli Maurer. Modelling a public-key infrastructure. In *European Symposium on Research in Computer Security*, pages 325–350. Springer, 1996.

[MBB+15] Marcela S Melara, Aaron Blankstein, Joseph Bonneau, Edward W Felten, and Michael J Freedman. Coniks: Bringing key transparency to end users. In *Usenix Security*, pages 383–398, 2015.

[moz] Mozilla included ca certificate list.

[MP07] Fabio Martinelli and Marinella Petrocchi. On relating and integrating two trust management frameworks. *Electronic Notes in Theoretical Computer Science*, 168:191–205, 2007.

[MS05] John Marchesini and Sean Smith. Modeling public key infrastructures in the real world. In *European Public Key Infrastructure Workshop*, pages 118–134. Springer, 2005.

[MT16] Michael Mitzenmacher and Manuel Torres. Models and algorithms for graph watermarking. In *Information Security: 19th International Conference, ISC 2016, Honolulu, HI, USA, September 3-6, 2016. Proceedings*, volume 9866, page 283. Springer, 2016.

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[nam] Namecoin.

[Res00] Eric Rescorla. Http over tls. 2000.

[SHYL06] Yan Lindsay Sun, Zhu Han, Wei Yu, and KJ Ray Liu. A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks. In *INFOCOM*, volume 2006, pages 1–13, 2006.

[Sle13] Greg Slepak. Dnschain+ okturtles, 2013.

[STV+16] Ewa Syta, Iulia Tamas, Dylan Visher, David Isaac Wolinsky, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ismail Khoffi, and Bryan Ford. Keeping authorities" honest or bust" with decentralized witness cosigning. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 526–545. IEEE, 2016.

[SY16] Kenji Saito and Hiroyuki Yamada. Whats so different about blockchain?blockchain is a probabilistic state machine. In *Distributed Computing Systems Workshops (ICDCSW), 2016 IEEE 36th International Conference on*, pages 168–175. IEEE, 2016.

[TD16] Alin Tomescu and Srinivas Devadas. Catena: Preventing lies with bitcoin. Cryptology ePrint Archive, Report 2016/1062, 2016. http://eprint.iacr.org/2016/1062.

[UHHC11] Alexander Ulrich, Ralph Holz, Peter Hauck, and Georg Carle. Investigating the openpgp web of trust. In *European Symposium on Research in Computer Security*, pages 489–507. Springer, 2011.

[VTM14] Marie Vasek, Micah Thornton, and Tyler Moore. Empirical analysis of denial-of-service attacks in the bitcoin ecosystem. In *International Conference on Financial Cryptography and Data Security*, pages 57–71. Springer, 2014.

[WA15] Duane Wilson and Giuseppe Ateniese. From pretty good to great: enhancing pgp using bitcoin and the blockchain. In *International Conference on Network and System Security*, pages 368–375. Springer, 2015.

[Wag14] Andrew Wagner. Ensuring network scalibility: How to fight blockchain bloat. *Bitcoin Magazine*, 6, 2014.

[WAP08] Dan Wendlandt, David G Andersen, and Adrian Perrig. Perspectives: Improving ssh-style host authentication with multi-path probing. In *USENIX Annual Technical Conference*, volume 8, pages 321–334, 2008.

[WMLZ14] Dongxia Wang, Tim Muller, Yang Liu, and Jie Zhang. Towards robust and effective trust management for security: A survey. In *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 511–518. IEEE, 2014.

[Woo14] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151, 2014.

[Zim95] Philip R Zimmermann. *The official PGP user's guide*. MIT press, 1995.