A Comparison of Blockchain-based PKI Implementations

Clemens Brunner, Fabian Knirsch, Andreas Unterweger and Dominik Engel
Center for Secure Energy Informatics, Salzburg University of Applied Sciences, Urstein Süd 1, Puch bei Hallein, Austria

Keywords: Blockchain, PKI, Web of Trust, X.509.

Abstract: Blockchain technology has recently been proposed by many authors for decentralized key management in the context of Public Key Infrastructures (PKIs). Instead of relying on trusted key servers – centralized or decentralized –, the confirmation and revocation of keys is distributed over a multitude of participants. A pletheora of implementations exist, all of which rely on different properties of blockchains. In this paper, we motivate the most relevant properties of blockchains as well as PKI and how they are linked. Furthermore, we provide an overview of state-of-the-art blockchain-based PKI implementations and compare them with respect to these properties. While all analyzed implementations fullfil the basic requirements of PKIs, we find that (i) privacy is very often neglected; and (ii) only a small subset is evaluated with respect to both, complexity and cost. In order to provide a guideline for future blockchain-based PKI implementations, we conclude with a set of recommendations based on our findings.

1 INTRODUCTION

A Public Key Infrastructure (PKI) is the fundamental building block of many applications that rely on secure and reliable authentication, such as digital signatures and encryption for email, smart cards and network connections. A PKI ensures that a certain entity is bound to its public key, usually by relying on trusted key servers maintained by Certificate Authorities (CA) (Gutmann, 2002). These authorities issue a certificate for a domain or a person that publicly and verifiably binds this entity to a certain key. A common format for such certificates is X.509 (Housley et al., 2008), which is used for, e.g., TLS (Rescorla, 2018; Dierks and Rescorla, 2008). Traditional PKI setups are mostly centralized and – despite being well established – face some problems, such as malicious certificates that can remain undetected and allow attackers to act as a man in the middle (Yu and Ryan, 2017). Similarly, the revocation of keys relies on a centralized list maintained by a few entities only. This implies a significant amount of trust that is put into a relatively small number of CAs. In recent years, the misuse of trust has lead to distrusting certificates from certain CAs altogether (Kumar et al., 2018).

One approach towards more transparency in the process of managing certificates has been proposed by (Laurie et al., 2013) and is referred to as *log-based* PKIs. The proposed public log allows to audit CA activity for the process of issuing, managing and revoking certificates, but does not provide a fully decentralized approach. The concept of such a public log has been advanced by the advent of blockchain technology in recent years. Blockchain technology provides a means for a public, decentralized, tamper-proof, complete and available list of records. A large number of blockchain-based, decentralized theoretical approaches, e.g., (Axon, 2015; Alexopoulos et al., 2017; Longo et al., 2017; Karaarslan and Adiguzel, 2018; Orman, 2018), have been discussed. They aim at tackling the challenges of traditional PKIs. Implementations of proposed approaches come with different storage types, permission models and support for certificate formats.

In this paper, we introduce and combine the most relevant properties of both, blockchain and PKI. This paper furthermore provides a comparison of state-of-the-art blockchain-based PKI implementations with respect to the identified properties. It is found that some of these aspects are not sufficiently covered by currently available implementations. Some related recommendations are given as guidelines for future blockchain-based PKI implementations.

The rest of this paper is structured as follows: In Section 2, we briefly introduce the relevant features of blockchains as well as properties of PKI with respect to different trust models. From these features and properties, we derive a list of criteria which serves as the basis for a comparison of state-of-the-art blockchain-based PKI implementations in Section 3.

Brunner, C., Knirsch, F., Unterweger, A. and Engel, D.
A Comparison of Blockchain-based PKI Implementations.
DOI: 10.5220/0008914503330340
In *Proceedings of the 6th International Conference on Information Systems Security and Privacy (ICISSP 2020)*, pages 333-340
ISBN: 978-989-758-399-5; ISSN: 2184-4356

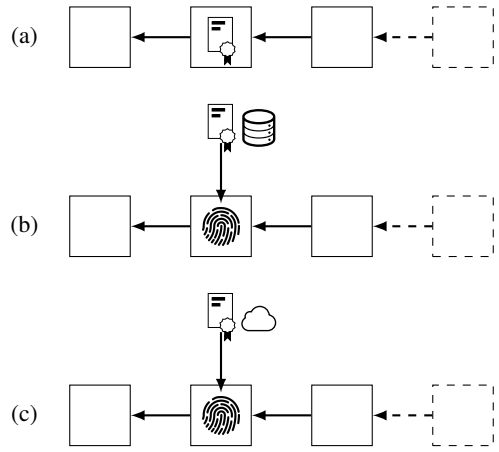

Figure 1: Three storage types are differentiated when building applications based on blockchain: (a) the full data (certificate icon) is stored on-chain; (b) only a hash reference (fingerprint icon) is stored on-chain, whereas the raw data is stored on a private server (database icon); (c) for the raw data, e.g., a DHT (cloud icon) is used.

In Section 4, we provide a summary and give recommendations before concluding the paper in Section 5.

2 BLOCKCHAIN AND PKI

In this section, we introduce both, blockchain and PKI, as well as their relevant properties. We further show how these properties are linked together.

2.1 Blockchain

Blockchains, which have been introduced in (Nakamoto, 2008), are decentralized, append-only databases of signed transactions and/or operations that yield a new globally consistent state (Tschorsch and Scheuermann, 2016). Numerous variations exist, both at the conceptual and the implementation level. The distinguishing properties of these variations that are relevant in the context of this work are (Wüst and Gervais, 2017):

Permission Type: While all data is accessible to all participants in public (permissionless) blockchains, e.g., (Nakamoto, 2008; Wood, 2017), data is only selectively accessible in (private) permissioned blockchains, e.g., (Karlsson et al., 2018)[1], at participant (group) level.

Blockchain Type: Blockchain implementations can either build on established and well investigated technologies, e.g., (Nakamoto, 2008; Wood, 2017), or start from scratch. In the former case, an existing public blockchain can be used, e.g., Ethereum, or a fork of an existing code base can be used to set up a new blockchain. In the latter case, security and end-user acceptance may be impacted negatively (Eyal and Sirer, 2014).

Storage Type: Data in blockchains can either be stored in full, e.g., (Nakamoto, 2008; Wood, 2017), or only through its hash, e.g., (Karaarslan and Adiguzel, 2018; Brunner et al., 2019). In the latter case, the actual data is stored off-chain, e.g., on a private server or in a Distributed Hash Table (DHT) like IPFS[2]. The different ways to store data on a blockchain are illustrated in Figure 1. The following storage categories exist:

- **On-chain:** Data relevant for the use case is stored directly on the blockchain, i.e., within blocks and/or transactions. *Full* on-chain storage means that all data is stored on the block-chain. Some implementations store the *hash only*, with the actual data being stored elsewhere.
- **Off-chain:** When data is not stored on the blockchain, it can either be stored publicly or privately. *Public* storage is when everyone has access to the data without restrictions, e.g., through a publicly accessible Web server. *Private* storage allows only limited access and the data is kept under the control of a limited number of entities.
- **DHT:** A special form of off-chain storage is a *DHT*, where the stored data is distributed among multiple participants. A hash can be used to access the data. If a cryptographically secure hash function (Barker et al., 2012) is used within the DHT to address the stored data then the data can be timestamped, integrity and tramperproof protected by sealing the hashes in the blockchain (Brunner et al., 2019).

Privacy: Storing only references to data instead of the full data allows for privacy, if hashes or salted hashes (Delmolino et al., 2016; Brunner et al., 2019) are used. While permissioned blockchains can control access to data, for making them effectively private, additional privacy-enhancing technologies (PETs) need to be used in permissionless blockchains, see e.g., (Unterweger et al., 2018). Storing unsalted hashes without PETs is not considered privacy-preserving in the context of this paper.

Evaluation: Operations, transmissions and storage on blockchains can be evaluated in terms of time/space complexity and in terms of (monetary) cost, e.g., (Hammi et al., 2018; Unterweger et al., 2018; Yakubov et al., 2018b):

[1] https://eos.io/

[2] https://ipfs.io

- **Complexity:** The computational complexity indicates how much computing power or memory is needed with a changing number of users, objects or other dependent variables.
- **Cost:** Different blockchain implementations charge different fees for transactions and/or the data volume contained within. For practical implementations, it is important to consider these costs per user, per object or per any other dependent variable.

If both aspects are evaluated, the practicality of an implementation and its suitability for a certain use case, e.g., PKIs, can be judged more thoroughly.

2.2 PKI

PKIs are protocols for binding a public key to a name, email address or identity of an individual for authentication, establishing a secure communication channel and for verifying the creator of signatures. PKIs are used in, e.g., TLS (Rescorla, 2018), and to encrypt and sign emails (Garfinkel, 1994). The types of PKIs can be distinguished through the design of the trust model.

In hierarchically structured trust models, CAs are allowed to issue certificates to participants. The CAs issue other CAs' certificates, building a chain of trust starting from so-called root CAs. Conversely, in a Web of Trust (WoT), all participants are equal and are allowed to issue certificates and thus confirm all other participants' public keys.

Log-based PKIs, e.g., (Laurie et al., 2013), are an extension to both, hierarchical and WoT-based PKIs. They need a public append-only database where certificates are registered before they are considered valid. In this concept, all certificates are publicized so that misbehavior can be detected quickly and denounced publicly.

In this paper, only blockchain-based PKI implementations are considered, all of which follow the principles of log-based PKIs (due to the nature of the blockchain), which are:

Revocation: A revocation invalidates a formerly valid issued certificate and is typically stored in a revocation list (Rivest, 1998) or managed by the Online Certificate Status Protocol (OCSP) (Galperin et al., 2000). Both approaches rely on trusted third parties in order to provide the complete and unmodified status of all certificates for checking whether or not a certificate is revoked.

Certificate Format: X.509 (Housley et al., 2008) is an established format for hierarchical certificates, whereas PGP (Zimmermann, 1995) is commonly used for certificates in a WoT (Menezes et al., 1996). X.509 standardizes the structure of both, intermediate certificates belonging to a CA and entity certificates. The standard additionally allows for custom extensions. However, this limits interoperability with existing applications.

PKI Type: As mentioned above, the two main PKI types are *Hierarchical* and *WoT*. In the hierarchical approach, a path from a valid certificate to a root CA always exists. Conversely, in a WoT, multiple paths of confirmations need to be checked and potentially rated for reliability (Caronni, 2000).

Incentives: Either rewards exist to incentivize honest behavior of CAs or penalties exist to discourage dishonest behavior (Matsumoto and Reischuk, 2016).

Updateable Key: When an issuer changes its key and revokes its previous key, issued certificates become invalid. If an update mechanism for the key exists, certificates may remain valid (Fromknecht et al., 2014).

2.3 Summary

The main properties of blockchain and PKI are largely independent of one another and complement each other for establishing a log-based PKI. A public and permanent log is maintained by means of a blockchain, removing the need for single trusted servers, e.g., for the purpose of maintaining revocation lists. State-of-the-art implementations based on this premise are evaluated in the next section.

3 COMPARISON OF PKI IMPLEMENTATIONS

The properties from Section 2 for both, blockchain and PKI are combined to serve as the basis for the comparison of state-of-the-art implementations which are available at the time of writing (September 2019). Proposals which do not provide a full implementation or a proof of concept, e.g., (Fromknecht et al., 2014; Axon and Goldsmith, 2017), are explicitly excluded in this paper to limit the scope. Furthermore, only peer-reviewed publications are taken into consideration.

Table 1 summarizes the properties of state-of-the-art blockchain-based PKI implementations. Templated approaches which require concrete blockchains instantiations, such as Hyperledger Fabric CA[3], are explicitly excluded. A short discussion of the general observations property by property follows.

[3]https://github.com/hyperledger/fabric-ca

Permission Type: The majority of implementations is permissionless, but nearly as many permissioned implementations exist. No clear trend can be observed. While permissionless implementations are to be preferred for Internet-scale applications such as domain name holder verification, permissioned approaches are better suited for small-scale use cases with a small number of participants.

Revocation: Revocation is explicitly built in by all but one implementation. However, the design of this implementation would allow for revocation logic despite not being mentioned in the paper. Revocation is one of the key security properties of PKIs and traditionally relies on trusted third parties to manage and distribute the revocation lists or the OCSP, respectively. The use of blockchains spreads this trust over all entities. Thus, it is positive that practically all approaches implement this property.

Blockchain Type: Most implementations are based on Ethereum. Some of these use the public Ethereum blockchain, while others set up a blockchain based on this technology. Other technology bases are used as well, including custom blockchains. One implementation (Ali et al., 2016) is built on Namecoin which itself is based on Bitcoin. One approach (Lei et al., 2019) only simulates the creation of blocks. Using an established public blockchain is beneficial in terms of reliability, tamper-proofness and availability. Conversely, when setting up a custom public blockchain, gathering support of participants is crucial for long-term stability and security.

Certificate Format: X.509 and custom, implementation-specific formats are the most common certificate formats. One implementation (Wilson and Ateniese, 2015), however, uses the PGP format (Zimmermann, 1995). The adherance to standards is desirable and simplifies interoperability with existing systems. While some implementations rely on extensions to X.509 as specified in X.509 v3, others use custom extensions to X.509 which do not facilitate interoperability.

PKI Type: The majority of implementations is hierarchical, but a considerable number of implementations is based on WoT. It should be noted that hierarchical PKIs imply the existence for root CAs which centralize trust. The properties of permissionless blockchains, however, allow for decentralizing trust. Using WoT instead of a hierarchical structured PKI is capable of eliminating these remaining trusted central components.

Storage Type: Full data storage on-chain as well as the storage of hashes are equally common. Only one implementation (Yakubov et al., 2018a) stores certificate data entirely off-chain.

The majority of implementations provide the certificate data (both, on- or off-chain) publicly. Only one implementation (Al-Bassam, 2017) uses a DHT to store the data off-chain. An exception is (Matsumoto and Reischuk, 2016), where only parts of the data are stored on-chain.

In (Ali et al., 2016), hashes are stored on-chain and the protocol allows for storing the corresponding certificate data either (i) privately; (ii) publicly without a DHT; or (iii) publicly with a DHT. Keeping the choice of data storage location flexible enables a multitude of applications on top of this work.

Updateable Key: Fewer than half of all implementations allow for updateable issuer keys. Some implementations would most likely support this feature with corresponding design changes. The ability to update keys is crucial for long-term applications. One approach to enable this is proposed in theoretical work (Fromknecht et al., 2014) and relies on two keys – an online key and an offline key. The former is used for all interactions until it is updated by the latter. It would be desirable that this feature be supported by all of the available implementations.

Privacy: No implementation explicitly takes privacy into consideration. While classical PKIs in general are not necessarily required to consider privacy, -based PKIs pose additional challenges, e.g., fully disclosing all certificate data, penalties (if applicable) etc. Permissioned blockchains limit read access, but do not provide privacy to a full extent. It would be desirable if any of the available implementations would make use of the large number of existing PETs, e.g., (Ben-Sasson et al., 2014; Delmolino et al., 2016; Knirsch et al., 2017).

Incentives: Only four implementations provide either incentives or penalties for (dis-)honest behavior. Most implementations, however, do not cover this aspect. Many implementations provide incentives for participants via built-in tokens. These tokens can be used to additionally incentivize good behavior in the context of PKIs. Blockchain-based approaches allow for a transparent and decentralized tracking of such reputation values. Such an automatic and trust-less mechanism for managing the reputation among participants would be desirable.

Evaluation: Only three implementations perform an evaluation of both, complexity and cost. Two implementations perform no evaluation. The rest evaluate only one of the two. Evaluating complexity and cost is highly relevant for blockchain-based PKIs. Thus, it would of great value if all implementations perform both evaluations to show their practicability and to make comparing them easier.

In summary, a large variety of implementations exist. They differ in practically all of the relevant aspects with no clear trend towards a unified standard being observable.

4 SUMMARY AND RECOMMENDATIONS

The analysis in Section 3 revealed that no state-of-the-art implementation covers all relevant aspects. We recommend that any blockchain-based PKI implementation supports at least the following aspects:

Permission Type: Both, public and permissioned blockchains can be used, depending on the use case and the number of participants (Knirsch et al., 2019). However, for most applications that implement large-scale generally accessible PKIs, public blockchains are desirable to to enhanced stability, security and end-user acceptance.

Revocation: Support for revoking certificates in a PKI is a must. Revocation is a key feature of PKIs as this is the only way to ensure the long-term validity of a certain certificate.

Blockchain Type: A public, existing, broadly backed and well-studied blockchains should be preferred over custom implementations due to stability and transparency.

Certificate Format: Support for an established and standardized certificate format with a minimum number of custom extensions is highly recommended.

PKI Type: For decentralized use cases, WoT is recommended, while domain-specific use cases may benefit from hierarchical PKIs. The former is a more decentralized and better reflecting the essence of blockchain-based approaches.

Storage Type: Only a minimum amount of data should be stored on the blockchain due to costs and performance reasons. If hashes are stored on the blockchain, a DHT is recommended for distributed off-chain storage.

Updatable Key: Support for updating issuer keys in a PKI is a must for long-term applications. Otherwise, certificates may prematurely become invalid due to a key update from the issuer.

Privacy: The need and use of privacy-enhancing technologies depends on the use case. For example, if the PKI links individuals to personal identifiable data (European Parliament and Council of the European Union, 2016) it is a legal requirement to apply appropriate privacy measures, e.g., (Brunner et al., 2019). In other cases, however, where certificate transparency, e.g., (Laurie et al., 2013) is desired, privacy by design is not necessarily required. In all other cases, privacy by design must be considered and appropriate privacy-enhancing technologies must be chosen.

Incentives: On custom blockchains, incentives for participants are necessary to establish a stable infrastructure. Conversely, the support for PKIs based on public blockchains is beneficial, but not mandatory.

Evaluation: Conducting a thorough analysis of both, the time and space complexity as well as the monetary cost, is a must for reproducability and comparability. Having a common methodology would greatly benefit the comparison of different PKI implementations.

In summary, the two most relevant shortcomings identified in the literature review are a lack of privacy and a lack of evaluation. As stated above, privacy is essential due to legal requirements in use cases involving individuals covered by data protection legislature. Similarly, a common evaluation methodology for both, cost and complexity is essential for comparing different approaches.

Furthermore, it would be highly desirable to have a common standard for blockchain-based PKI implementations, as it is currently drafted for blockchain identity management systems (Lesavre et al., 2019).

5 CONCLUSION AND OUTLOOK

Using blockchains for PKIs is promising. A number of implementations exist already, most of which cover important aspects, such as certificate revocation. However, there are two main shortcomings to be observed in current implementations: First, no implementation explicitly considers privacy-enhancing technologies. Existing solutions are based on permissions and the storage of hashes instead of full certificate data. Second, only two implementations analyze both, complexity and cost. To evaluate and compare implementations, it is highly desirable that more papers analyze these aspects and do so in a standardized fashion.

In summary, it is possible to remove central trusted third parties with state-of-the-art blockchain-based PKIs. Some directions for future research and potential solutions have been hinted, but implementations solving all of these open issues remain future work.

Table 1: Comparison of blockchain-based PKIs based on the criteria from Section 2. * denotes a standardized certificate format with a custom extension.

Paper	Permission Type (Is Permissionless)	Revocation	Blockchain Type	Certificate Format	PKI Type	Storage Type			Updatable Key	Privacy	Incentives	Evaluation	
						On-chain	Off-chain	DHT				Complexity	Cost
(Khieu and Moh, 2019)	✓	✓	Ethereum	X.509	Hierarchical	Hash only	Private	✗	✗	✗	✗	✓	✓
(Viriyasitavat et al., 2019)	✗	✓	Custom	Custom	Hierarchical	Full	Private	✗	✗	✗	✗	✓	✗
(Lei et al., 2019)	✗	✓	N/A	X.509*	Hierarchical	Full	Private	✗	✓	✗	✗	✓	✗
(Kubilay et al., 2019)	✗	✓	Ethereum-based	X.509	Hierarchical	Hash only	Private	✗	✗	✗	✓	✓	✗
(Ahmed and Aura, 2018)	✓	✓	Ethereum	X.509 v3	Hierarchical	Full	Public	✗	✗	✗	✗	✗	✓
(Hammi et al., 2018)	✗	N/A	Ethereum-based	Custom	WoT	Full	Public	✗	✗	✗	✗	✓	✓
(Chen et al., 2018)	✗	✓	Ethereum	Custom	WoT	Hash only	Private	✗	✓	✗	✓	✓	✗
(Yakubov et al., 2018a)	✗	✓	Ethereum	X.509*	WoT	None	Public	✗	✓	✗	✗	✗	✗
(Yakubov et al., 2018b)	✓	✓	Ethereum	X.509 v3	Hierarchical	Full	Public	✗	✗	✗	✗	✓	✓
(Wang et al., 2018b)	✓	✓	Custom	X.509	Hierarchical	Full	Public	✗	✓	✗	✗	✓	✗
(Wang et al., 2018a)	✗	✓	Custom	X.509*	Hierarchical	Hash only	Private	✗	✓	✗	✗	✓	✗
(Al-Bassam, 2017)	✓	✓	Ethereum	Custom	WoT	Hash only	Public	✓	✗	✗	✗	✗	✓
(Matsumoto and Reischuk, 2016)	✓	✓	Ethereum-based	X.509	Hierarchical	Partial	Public	✗	✓	✗	✓	✗	✓
(Ali et al., 2016)	✓	✓	Namecoin	Custom	WoT	Hash only	Multiple options		✗	✗	✗	✓	✗
(Wilson and Ateniese, 2015)	✓	✓	Bitcoin	PGP*	WoT	Full	Public	✗	✗	✗	✓	✗	✗

ACKNOWLEDGEMENTS

The financial support by the Federal State of Salzburg is gratefully acknowledged. Funding by the Austrian Research Promotion Agency (FFG) under project number 865082 (ProChain) is gratefully acknowledged.

REFERENCES

Ahmed, A. S. and Aura, T. (2018). Turning Trust Around: Smart Contract-Assisted Public Key Infrastructure. In *17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pages 104–111, New York, NY, USA. IEEE.

Al-Bassam, M. (2017). SCPKI: A Smart Contract-based PKI and Identity System. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, pages 35–40, Abu Dhabi, UAE. ACM.

Alexopoulos, N., Daubert, J., Muhlhauser, M., and Habib, S. M. (2017). Beyond the Hype: On Using Blockchains in Trust Management for Authentication. In *2017 IEEE Trustcom/BigDataSE/ICESS*, pages 546–553, Sydney, Australia. IEEE.

Ali, M., Nelson, J., Shea, R., and Freedman, M. J. (2016). Blockstack: A Global Naming and Storage System Secured by Blockchains. In *USENIX Annual Technical Conference*, pages 181–194, Denver. USENIX Association.

Axon, L. (2015). Privacy-awareness in blockchain-based PKI. Technical report. http://ora.ox.ac.uk/objects/uuid:f8377b69-599b-4cae-8df0-f0cded53e63b.

Axon, L. and Goldsmith, M. (2017). PB-PKI : a Privacy-Aware Blockchain-Based PKI Conventional Approaches to PKI. In *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017)*, pages 311 —- 318, Madrid, Spain.

Barker, E., Barker, W., Burr, W., Polk, W., Smid, M., and Division, C. S. (2012). NIST 800-57: Computer Security.

Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., and Virza, M. (2014). Zerocash: Decentralized Anonymous Payments from Bitcoin. In *Proceedings – IEEE Symposium on Security and Privacy*, pages 459–474, San Jose, CA, USA. IEEE.

Brunner, C., Knirsch, F., and Engel, D. (2019). SPROOF: A platform for issuing and verifying documents in a public blockchain. In *Proceedings of the 5th International Conference on Information Systems Security and Privacy*, pages 15–25, Prague, Czech Republic. SciTePress.

Caronni, G. (2000). Walking the Web of Trust. In *9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2000)*, pages 153–158, Gaithersburg, MD, USA. IEEE.

Chen, J., Yao, S., Yuan, Q., He, K., Ji, S., and Du, R. (2018). CertChain: Public and Efficient Certificate Audit Based on Blockchain for TLS Connections. In *IEEE International Conference on Computer Communications*, pages 2060–2068, Honolulu, HI, USA. IEEE.

Delmolino, K., Arnett, M., Kosba, A. E., Miller, A., and Shi, E. (2016). Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab. In *Financial Cryptography and Data Security*, pages 79–94, Christ Church, Barbados. Springer, Berlin, Heidelberg.

Dierks, T. and Rescorla, E. (2008). The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246. https://tools.ietf.org/html/rfc5246.

European Parliament and Council of the European Union (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Da. http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679.

Eyal, I. and Sirer, E. G. (2014). Majority is not Enough: Bitcoin Mining is Vulnerable. In *Financial Cryptography and Data Security*, pages 436–454, Christ Church. ACM.

Fromknecht, C., Velicanu, D., and Yakoubov, S. (2014). CertCoin: A NameCoin Based Decentralized Authentication System. Technical report, MIT, Cambridge, MA, USA. https://courses.csail.mit.edu/6.857/2014/files/19-fromknecht-velicann-yakoubov-certcoin.pdf.

Galperin, S., Santesson, S., Myers, M., Malpani, A., and Adams, C. (2000). X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC2560. https://tools.ietf.org/html/rfc2560.

Garfinkel, S. (1994). *PGP: Pretty Good Privacy*. O'Reilly Media, Sebastopol, CA, USA.

Gutmann, P. (2002). PKI: It's not dead, just resting. *Computer*, 35(8):41–49.

Hammi, M. T., Bellot, P., and Serhrouchni, A. (2018). BC-Trust : A decentralized authentication blockchain-based mechanism. In *IEEE Wireless Communications and Networking Conference*, pages 1–6, Barcelona, Spain. IEEE.

Housley, R., Polk, W., Ford, W., and Solo, D. (2008). Internet X. 509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280. https://tools.ietf.org/html/rfc5280.

Karaarslan, E. and Adiguzel, E. (2018). Blockchain Based DNS and PKI Solutions. *IEEE Communications Standards Magazine*, 2(3):52–57.

Karlsson, K., Jiang, W., Wicker, S., Adams, D., Ma, E., Van Renesse, R., and Weatherspoon, H. (2018). Vegvisir: A partition-tolerant blockchain for the internet-of-things. In *International Conference on Distributed Computing Systems*, pages 1150–1158. IEEE.

Khieu, B. and Moh, M. (2019). CBPKI : Cloud Blockchain-based Public Key Infrastructure. In *ACM SE '19: Proceedings of the 2019 ACM Southeast Conference*, pages 58–63, Kennesaw, GA, USA. ACM.

Knirsch, F., Eibl, G., and Engel, D. (2017). Multi-resolution privacy-enhancing technologies for smart metering. *Eurasip Journal on Information Security*, 2017(1):1–13.

Knirsch, F., Unterweger, A., and Engel, D. (2019). Implementing a Blockchain from Scratch: Why, How, and What We Learned. *EURASIP Journal on Information Security*, 2019(2):1–14.

Kubilay, M. Y., Kiraz, M. S., and Mantar, H. A. (2019). CertLedger: A new PKI model with Certificate Transparency based on blockchain. *Computers and Security*, 85:333–352.

Kumar, D., Wang, Z., Hyder, M., Dickinson, J., Beck, G., Adrian, D., Mason, J., Durumeric, Z., Halderman, J. A., and Bailey, M. (2018). Tracking Certificate Misissuance in the Wild. In *Proceedings - IEEE Symposium on Security and Privacy*, pages 785–798. IEEE.

Laurie, B., Langley, A., and Kasper, E. (2013). Certificate Transparency. RFC 6962. https://tools.ietf.org/html/rfc6962.

Lei, A., Cao, Y., Bao, S., Li, D., Asuquo, P., Cruickshank, H., and Sun, Z. (2019). A blockchain based certificate revocation scheme for vehicular communication systems. *Future Generation Computer Systems*, (in press).

Lesavre, L., Varin, P., Mell, P., Davidson, M., and Shook, J. (2019). A Taxonomic Approach to Understanding Emerging Blockchain Identity Management Systems. Technical report, National Istitute of Standards and Technology. https://doi.org/10.6028/NIST.CSWP.07092019-draft.

Longo, R., Pintore, F., Rinaldo, G., and Sala, M. (2017). On the security of the blockchain BIX protocol and certificates. In *International Conference on Cyber Conflict, CYCON*, pages 1–16, Tallinn, Estonia. IEEE.

Matsumoto, S. and Reischuk, R. (2016). IKP: Turning a PKI Around with Blockchains. Technical report. https://eprint.iacr.org/2016/1018.pdf.

Menezes, A. J., Van Oorschot, P. C., and Vanstone, S. A. (1996). *Handbook of applied cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1 edition.

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Technical report. https://bitcoin.org/bitcoin.pdf.

Orman, H. (2018). Blockchain: The Emperor's New PKI? In *IEEE Internet Computing*, volume 22, pages 23–28. IEEE.

Rescorla, E. (2018). The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446. https://tools.ietf.org/html/rfc8446.

Rivest, R. L. (1998). Can we eliminate certificate revocation lists? In *International Conference on Financial Cryptography*, pages 178–183, Anguilla. Springer, Berlin, Heidelberg.

Tschorsch, F. and Scheuermann, B. (2016). Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Communications Surveys \& Tutorials*, 18(3):2084–2123.

Unterweger, A., Knirsch, F., Leixnering, C., and Engel, D. (2018). Lessons Learned from Implementing a Privacy-Preserving Smart Contract in Ethereum. In *9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5, Paris, France. IEEE.

Viriyasitavat, W., Xu, L. D., Bi, Z., and Sapsomboon, A. (2019). New Blockchain-Based Architecture for Service Interoperations in Internet of Things. In *IEEE Transactions on Computational Social Systems*, volume 6, pages 1–10. IEEE.

Wang, W., Hu, N., and Liu, X. (2018a). BlockCAM: A blockchain-based Cross-domain Authentication Model. In *IEEE 3rd International Conference on Data Science in Cyberspace*, pages 896–901, Guangzhou, China. IEEE.

Wang, Z., Lin, J., Cai, Q., Wang, Q., Jing, J., and Zha, D. (2018b). Blockchain-based Certificate Transparency and Revocation Transparency. In *Financial Cryptography and Data Security*, Nieuwpoort, Curaçao. Springer, Berlin, Heidelberg.

Wilson, D. and Ateniese, G. (2015). From pretty good to great: Enhancing PGP using bitcoin and the blockchain. In *International Conference on Network and System Security*, pages 368–375. Springer.

Wood, G. (2017). Ethereum: A Secure Decentralised Generalised Transaction Ledger. Technical report, Ethereum. https://ethereum.github.io/yellowpaper/paper.pdf.

Wüst, K. and Gervais, A. (2017). Do you need a Blockchain. Technical report, International Association for Cryptologic Research. https://eprint.iacr.org/2017/375.pdf.

Yakubov, A., Shbair, W., and State, R. (2018a). BlockPGP: A Blockchain-Based Framework for PGP Key Servers. In *2018 Sixth International Symposium on Computing and Networking Workshops (CANDARW)*, pages 316–322, Takayama, Japan. IEEE.

Yakubov, A., Shbair, W. M., Wallbom, A., Sanda, D., and State, R. (2018b). A Blockchain-Based PKI Management Framework. In *The First IEEE/IFIP International Workshop on Managing and Managed by Blockchain (Man2Block) colocated with IEEE/IFIP NOMS*, pages 1–6, Tapei, Tawain. IEEE.

Yu, J. and Ryan, M. (2017). Evaluating web PKIs. In *Software Architecture for Big Data and the Cloud*, chapter 7, pages 105–126. Elsevier.

Zimmermann, P. R. (1995). *The Official PGP User's Guide*. MIT Press, Cambridge, MA, USA.