Preprint from https://www.gipp.com/pub/

T. Hepp et al. "Exploring Potentials and Challenges of Blockchain-based Public Key Infrastructures". In: *IEEE INFOCOM 2019 - 2nd Workshop on on Cryptocurrencies and Blockchains for Distributed Systems (CryBlock 2019).* Paris, France, May 2019

Exploring Potentials and Challenges of Blockchain-based Public Key Infrastructures

Thomas Hepp[1], Fabian Spaeh[1], Alexander Schoenhals[1], Philip Ehret[1], and Bela Gipp[2]

[1]Department of Computer and Information Science, University of Konstanz
[2]Chair of Media Technology, University of Wuppertal

***Abstract*—Traditional public key infrastructures (PKIs), in particular, X.509 and PGP, is plagued by security and usability issues. As reoccurring incidents show, these are not only of theoretical nature but allow attackers to inflict severe damage. Emerging blockchain technology allows for advances in this area, facilitating a trustless immutable ledger with fast consensus. There have been numerous proposals for utilization of the blockchain in the area of PKI, either as extensions upon existing methods or independent solutions. In this paper, we first study traditional PKI, then proceed with novel approaches, showing how they can improve upon recent issues. We provide a comprehensive evaluation, finding that independent blockchain-based solutions are preferable in the future, mainly due to their stronger security. However, global adoption of these yet requires advances in blockchain development, e.g., concerning scalability.**

***Index Terms*—Blockchain, Public Key Infrastructure, Web of Trust**

I. Introduction

Secure communication over an insecure network can only be guaranteed if there exists a way to verify the other party's identity. Otherwise, an attacker can easily intercept the communication and impersonate involved entities, e.g., by performing a man-in-the-middle attack. After the introduction of public key cryptography, the challenge transformed into the decision making whether a partner's public key can be trusted. "Trusted", in this context, means that only the intended partner in communication holds the respective private key. If the two communicating partners share a shared secret, i.e., a passphrase, the solution becomes trivial. However, as sharing secrets on a global scale is impractical, more sophisticated techniques need to be employed - all collected under the term Public Key Infrastructure (PKI). Two popular solutions nowadays are X.509 and PGP. Differing in design, X.509 is used to secure communication with websites, whereas PGP is prevalent in the field of securing emails. X.509 relies heavily on trusted third parties, giving rise to numerous security failures [1], e.g., the Symantec incident [2]. But also PGP suffers from issues, foremost related to its bad usability [3]. A blockchain could be used to overcome the need for trusted third parties and help to solve problems related to PGP. There have been numerous proposals for blockchain-based PKI in the literature, leading to the question of whether they can outperform classical PKI and how they compare amongst each other.

In this paper, we indent to give a broad overview and detailed comparison of proposed ideas to evaluate their applicability and performance. PKI and related terms are introduced in Section II, including some evaluation criterion applied later. Section II covers PKI solutions which are currently deployed, along with some improvements. We contrast these in Section III with PKI schemes based on blockchain technology (BT). These are coarsely split into schemes improving upon existing PKIs, independent blockchain solutions, and ones based on designated blockchains. Finally, in Section IV, we compare and evaluate all introduced PKIs and conclude in Section V with an outlook.

II. Theoretical Background

In this chapter, we want to provide a background on PKI, beginning with restating our original motivation. We then name some reoccurring entities and conclude with possible threats and challenges used later in our evaluation. For the set of standard terms for PKI, we want to refer to Vacca [4] due to the limited scope of our paper.

A. Objectives

The following objectives for trusted communication between partners over an insecure channel are the fundament for the evolution of PKI.

Privacy. Only the intended communication partner can read encrypted messages.

Non-repudiation. Users can be held responsible for their actions, i.e., signed messages cannot be denied.

Integrity. No one can tamper with data in transit over an insecure network or during long-term storage.

Authentication Users can confirm their identities to an authority.

In the PKI model, the attacker tries to trick the relying party into trusting a forged certificate containing a wrong subject identifier. Attack vectors are further classified into the categories proposed by Alexopoulos et al. [5] and [6].

B. Challenges

Finally, we present common challenges and evaluation criteria for a PKI. These criteria are used in Section 4 to compare all introduced PKI designs.

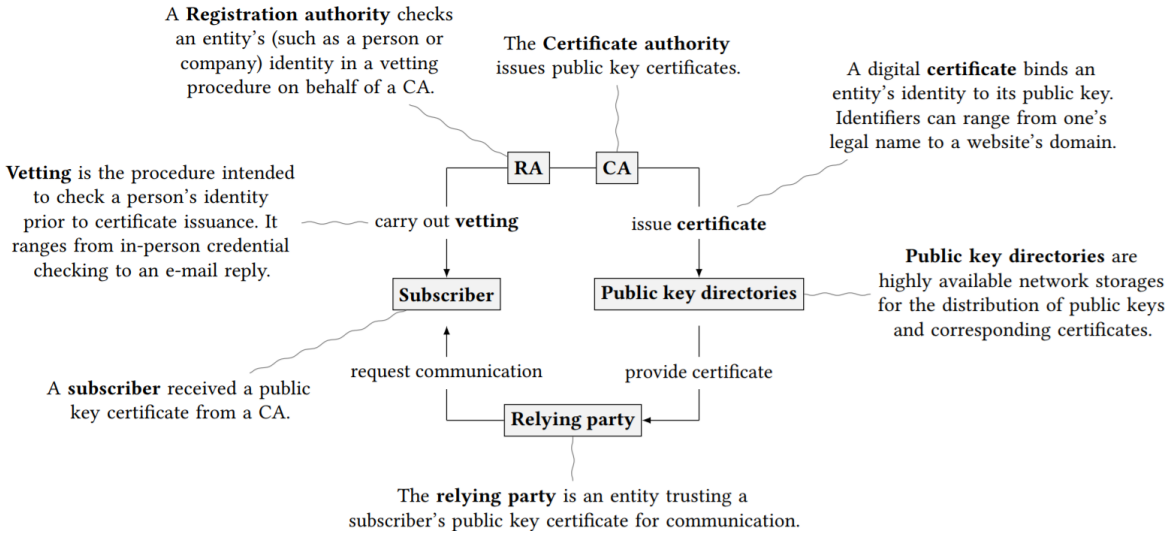

Fig. 1: Schematic overview of entities in PKI. Note, that in a real scheme, entities often concur. So does a CA always include its RA unless otherwise stated

Security Does the PKI mitigate all attacks mentioned above, to allow for privacy, non-repudiation, integrity, and authentication?

Scalability Does the architecture scale to a global level, e.g., to support safe browsing on millions of websites?

Client usability Can even inexperienced users securely operate and interact with the PKI user interface?

Speed How fast is certificate validation?

Flexibility To how many situations can the PKI be applied and how flexible are its policies during the certificate lifecycle?

Cost of operation How high are the costs to operate the infrastructure and how much of these costs are passed on to the subscriber or relying on the party?

C. X.509

X.509 [8] is intended to be an all-around solution applicable to many scenarios, but its primary application is securing the HTTPS protocol [9] as TLS.

A top-level (or root CA), is intended as a trust-anchor, from which trust is inherited. CAs sign certificates for every lower level of CA's public key. The lowest CA then signs a subject's certificate after successful vetting with an RA. If the relying party can validate this chain of signing CAs up to the root CA's certificate, the certificate is accepted as valid, and the connection is declared secure.

Key revocation becomes necessary after the key compromise of either any CA in the chain or the subscriber or only due to a change in the trust relationship. X.509 requires the higher-level CA to distribute information about revoked certificates, which is either done by certificate revocation lists (CRL) or upon request.

TLS [10] is a protocol applying X.509 to secure data transfer over the internet and is used inside web browsers as HTTPS. Regular X.509 certificates are extended to include either the domain name (domain-validated (DV) certificates) or additionally the name of a legal entity or company (extended validation (EV) certificates).

The X.509 standard is known to have multiple issues, many as a symptom of its design for broad applicability [11]:

Misbehaving CAs. Over one thousand CAs [12] in several countries with different legal systems exist. As any CA can issue a certificate for any domain, infiltrating a CA is very attractive for an attacker. Compromising a CA allows successful MITM attacks, like the *DigiNotar incident*.

Inconsistent naming. No agreement exists between CAs on how to create meaningful identifiers for PKI entities [11].

Key revocation. CRLs are refreshed in a fixed interval, in between which an attacker can inflict a lot of damage. OCSP requires checks on every request and is questionable in terms of privacy, as the OCSP server knows about every website browsed.

D. PGP

PGP is a program suite intended to facilitate secure communication [13]. It refrains from using authorities. Instead, users act as CA and subscriber simultaneously. PGP also includes techniques for encryption. However, we focus on how PGP and its superordinate standard OpenPGP [14] implements an infrastructure of trust.

As a user's identity typically receives multiple signatures and trust has to be evaluated on a per-user basis, we speak about trust and validity in a user's identity. PGP does not impose any structure on certificate signing, in particular, no hierarchical as X.509 does. Instead, it aims to create a web of trust: Trust relationships expressed as signatures can occur unconstrained. To establish a single trust relationship, user S (acting as the subscriber) has to undergo a vetting procedure conducted by another user C

(acting as the CA). After successful credential checking, user C signs all provided information together with S's public key and thereby attests trust in this certificate. To reduce the number of direct links needed in the web of trust, but keep it challenging for an attacker to be deeply incorporated, PGP makes use of validity levels for identities. The most common way (as e.g. employed by GnuPG [15]) is based on trust levels assigned to other users. Other methods assign values on every link in the web and compute validity levels in the form of probabilities [16]. Several key servers are in place to distribute the public keys and certificates of the users. Servers in the pool synchronize their entries amongst each other but make it an individual decision on whether to include a particular entry [17]. Users can download certificates on demand and keep these locally in a keyring. Revocation as a result of private key loss or withdrawal from a trust relationship is carried out by uploading a signed revocation certificate to a key server. The server then removes the certificate in question from its database. Although widely adopted by advanced users, PGP is still facing the following issues.

Usability. PGP clients lack easy usability by average users [3] and keyring management is complicated. **Certificate distribution.** Key servers are not required to share a consensus which allows an attacker to run his key server. Certificate revocation from a single key server takes time to distribute and may not be carried out by other key servers at all. **Scalability.** Identity certificate signing is not incentivized and hence, incorporation of one's certificate in the web of trust consumes time and in-person vetting with other users.

III. Blockchain-based PKI

The blockchain can be understood as a distributed ledger with fast consensus and no need for trusted third parties. Looking back to the issues of previously discussed PKIs, these properties seem desirable.

A. Blockchain backed X.509

One way of combining X.509 certificates and BT is by appending every certificate to the blockchain. X.509's hierarchical structure is replicated on the blockchain in smart contracts, such that its issuer always controls a certificate. However, revocation rights are also granted directly to the subscriber. CA misbehavior is tracked on the blockchain and can be noticed by all participating entities. At the same time, identity retention is enforced by design. As certificate revocation is now in the hands of the subscriber, there is no need to make a detour over the CA. Moreover, due to a blockchain's fast consensus, revocation information is spread instantly across the network. In the implementation of Yakubov et al. [18], an X.509 certificate's extension field is used to indicate its location in the blockchain. Smart contracts for each CA contain one list with all issued certificates and another list for revoked certificates. A valid certificate must hence prove its absence from the respective revocation list.

B. Blockchain-based PGP

Wilson and Ateniese [19] describe how to complement the current PGP infrastructure with Bitcoin's blockchain. Certificate generation, signing, and revocation are all implemented as transactions. A Bitcoin address extends identity certificates. The signer sends funds to the certificate holder's address, who in turn is required to send these back. The amount transferred by the signee reflects trust in identity. Afterward, the signers are granted a small endorsement fee in return for their efforts. Certificate revocation is performed directly by the certificate owner. Validation is carried out as usual by assigning validity levels, but individual links in the web of trust can be assigned a value proportional to the amount that was shifted back and forth during vetting. Trust relationships are assigned a real monetary value, and user's signing histories can be fully traced on the blockchain. The blockchain enforces consensus amongst all certificates, enabling identity retention and sound certificate revocation. BT is more potent against network attacks and censorship as a traditional key server. Furthermore, the endorsement fee might act as an incentive leading to increased adoption of PGP. The implementation of Wilson and Ateniese [19] further includes a key server which connects the user to the blockchain through a common PGP interface.

C. Decentralized public-key infrastructure (DPKI)

DPKI is not an implementation of a PKI, but merely a set of requirements on how to implement identity management in a decentralized manner [20]. The basic idea is to give full control over an identifier to its owner. Identifiers are placed in specific namespaces, each of which has custom rules for registration, expiration, and renewal. Two kinds of keys are used: A master key protecting the identifier itself, and subkeys, i.e., public keys associated with the identifier. The master key has to be generated in a decentralized manner and can be shared across trusted parties to allow for recovery. Ethereum Name Service (ENS) is one DPKI compatible PKI implementation on the Ethereum blockchain. The administration is divided into one central smart contract, the ENS registry, and two other kinds of smart contracts, namely registrars and resolvers. The registry contains the list of all domains, each pointing to a resolver. A resolver then provides information about the domain, e.g., IP address or the public key used. Registrars are in place to manage subdomains. The system is flexible as smart contracts entirely control it. Revocation is instant, as a result of owners being in full control of their identifiers. To subvert it, an attacker would now not only have to compromise a single authority but attack multiple miners at once. Also, security issues plaguing DNS, which were intended to be solved by extensions such as DNSSEC [21], can be addressed all along. Note that ENS can be extended to support EV certificates by handing control of a registrar to a CA. Other DPKI compatible implementations are DIDs [22] and uPort.

D. Designated blockchains

Particular purpose blockchains typically offer less security and flexibility but compensate these drawbacks with better scalability and thin client protocols. *Namecoin* is a Bitcoin fork with the additional ability to store values associated with a key. After paying a fixed registration fee, a domain stays registered for around 200 days. Public keys can be added to a domain to enable PKI. Like ENS, Namecoin acts as DNS (for .bit domains) and PKI at the same time. The decreased flexibility compared to ENS is traded for better support of thin clients.

Permissioned blockchains. Fredriksson [23] proposes a blockchain solely designated to the management of PKI. The blockchain uses the proof-of-stake consensus, where stakeholders are a small group of semi-trusted entities, such as governments or CAs. Adding an identifier to the blockchain requires the signature of a participating CA, granted after a vetting procedure, and the signature of the identity holder itself. This process emphasizes the benefits of EV certificates, i.e., certificates associated with a legal entity, but identities are at the same time controlled by their respective identity holders. Permissioned blockchains can offer higher transaction throughput as their consensus protocol can be much simpler [24].

IV. Discussion

To simplify our discussion, we group similar approaches in the categories below. The discussion is summarized in Table I.

X.509 The X.509 infrastructure extended by Certificate Transparency and public key pinning or OCSP stapling.
PGP Plain PGP using key servers.
X.509 + B X.509 enhanced by backing certificates on the blockchain.
PGP + B PGP using the blockchain as key server and for meaningful links in the web of trust.
ENS Systems fully operating on the blockchain in accordance with DPKI, as exemplified by ENS
NC PKI running as a designated blockchain, such as Namecoin

A. Security

Proper vetting. Improper vetting is a problem in X.509, which is amplified by the high number of available CAs. Involving a blockchain cannot eliminate this issue, as certificates are only tracked after vetting. Being well incorporated into PGP's web of trust requires multiple signatures, hence more vetting by different users. The procedure is further secured by shifting funds to express trust in an identity. Both, ENS and NC, include a DNS right away and do not require vetting at all.

Identity retention. Double registration attacks are possible in X.509 as the protocol does not require consensus between CAs. With Certificate Transparency, it takes time to detect and revoke suspicious certificates. Also, PGP has no mechanism to enforce consensus, key servers can decide on certificates. This fact makes it possible to create a double certificate for any identity and upload it to a colluding key server. However, the attacker needs to conspire with a sufficiently high number of trusted users to succeed. By enforcing consensus, all blockchain-based solutions are inherently immune to double registration attacks.

Sound revocation. Certificate revocation is a slow process in X.509 as the CA controls revocation. Revocation of PGP certificates requires reliable key servers and users updating their key rings. Also, the cancellation of a trust relationship is not incentivized and mostly not carried out. All other blockchain-based approaches can make use of instant revocation of their certificate on the blockchain. The transaction containing a revocation only has to reach sufficient block depth to be recognized as genuine.

Censorship circumvention. Censorship can affect all stages of a certificate's lifecycle: Registration, renewal, and revocation. Due to its decentralized nature, registration is not severely affected by censorship in X.509, as an attacker would have to block access to all available CAs. In PGP, it is possible for an attacker to deny access to the key server network Hence all stages can be affected. A blockchain is unaffected by censorship [20], and so are blockchain-based approaches. Special purpose blockchains are more prone to centralization. X.509 + B might still be attacked depending on the need for interaction with a CA.

Operational security. Secure operation of X.509 CAs is not directly incentivized. PGP relies on the secure behavior of every user. But to incorporate a fake certificate deeply in the web of trust, it is necessary to subvert multiple users. Integrated blockchain-based approaches incentivize secure operation, as every failure is recorded publicly and irrevocably on the blockchain. Pure blockchain-based approaches are unaffected by such threats as long as the underlying blockchain is, which depends on the number of miners.

Transparency. Trust in X.509 is handed down over multiple CAs to a user, but the assertion made by such a trust relationship is unclear. PGP does not have liabilities, and its operation builds on honest users. Chains of trust are non-transparent, and there is no natural way to do trust computations. As the evaluation of the trustworthiness of CAs or users can be improved by using a blockchain as a public log due to its immutable transparency. Purely blockchain-based schemes do not need to use trust relationships at all, as they combine issuance of an identifier with the administration of public keys.

B. Scalability

There's no question of the scalability of X.509, as it currently operates HTTPS. Maintaining a PGP certificate, on the other hand, requires meeting multiple other certificate holders and execution of a vetting procedure. All blockchain-based approaches inherit the scalability problems of current blockchain solutions concerning transaction throughput. We briefly want to analyze the feasibility of

TABLE I: Comparison of PKIs

putting a PKI on the blockchain with the example of ENS in a simplified calculation:

1,222,845 transactions are recorded for the ENS smart contract, with 228,147 finalized domain auctions (3,863 of which on the busiest day), which amounts to roughly 5 auctions needed for acquisition and administration of a domain [25]. ENS has been in operation for 460 days since May 9, 2017. This results in an average number of

$$\frac{1,222,845\ t}{460\ days} \approx 0.03\ t/s \tag{1}$$

but up to

$$\frac{3,863 \cdot 5\ t}{1\ day} \approx 2\ t/s \tag{2}$$

on the busiest day [26].

The number of certificates found in Certificate Transparency logs gives a lower bound on the total number of SSL certificates and amounts to 2,237,614,184 [27]. The logs can give an estimate for newly registered certificates as well. As such reports, the Google Rocketeer logs 364,798,762 certificates since May 9, 2017 [28]. Ethereum currently supports 7-15 t/s, but developers estimate sharding and 2nd layer solutions to increase the throughput to 10,000 t/s [29]. Transferring the whole SSL infrastructure to ENS would, therefore, take at least

$$\frac{2,237,614,184 \cdot 5\ t}{15\ t/s} \approx 23\ years \tag{3}$$

with the currently possible transaction throughput. Supporting only current certificate registrations would require support for

$$\frac{364,798,762 \cdot 5\ t}{460\ days} \approx 46\ t/s \tag{4}$$

but substantially more during peak times, which is both impossible to process. However, the expected throughput of future Ethereum implementations is able to handle this.

C. Client usability

X.509 typically does not require user interaction for HTTPS; all that is done is indicating the presence of a website's certificate. The same holds for X.509 + B, ENS, and NC. Using a PGP client on the contrary, which includes administration of one's keyring and assignment of individual trust levels, requires trained and experienced users [3]. Involving a blockchain cannot help to simplify the process.

D. Speed

X.509 certificates can be transmitted alongside a request's response, and hence do not require large communication overhead. However, validating revocation can involve exponentially many requests to certificate authorities [11] when not using OCSP stapling. As X.509 + B does not require revocation lists, the certificate can be downloaded using a thin client protocol in parallel with the request. ENS and NC combine DNS with delivering certificates. Hence, looking up the IP address and public key can be combined. However, client protocols require querying the network multiple times to mitigate the risk of bogus responses. PGP is not intended to deliver trust information in real time, and an unseen certificate must be handled by the user first.

E. Flexibility

X.509 and PGP are already quite well applicable in a wide range of environments as a result of their signing structure. But, ENS and other PKI implemented as smart contracts offer the highest flexibility and can even comprise DNS. Namecoin does not support Turing complete smart contracts and registration cost and expiration time are fixed.

F. Cost of operation

CAs have to be paid for their services. Even though free services exist, they can only provide DV certificates. PGP is based on the voluntary creation of trust relationships, therefore free. Adding a blockchain does not increase costs for subscribers or users, but even creates funds for miners by operating the ledger.

V. Conclusion

We began this paper by introducing the two well-established PKIs X.509 and PGP. Both suffer from security issues, especially so X.509 due to its need for CAs. Its bad usability cumbers PGP. On the blockchain, it is possible to administrate a trustless and consistent database of records, suited to overcome issues with both PKIs. We explored

how different systems involve this new technology and how that affects security and other properties. Extensions to existing PKIs do not require updates of existing infrastructure but still, drag along trust issues. Purely blockchain-based solutions, on the other hand, offer better security and higher flexibility. Designated approaches gain better support for clients and scale better while possibly losing decentralization. Concluding the discussion, we give an outlook on the future of PKI. The popularity of Let's Encrypt shows that there is no need for in-person vetting in many cases, and so are fully blockchain-operated solutions like ENS favorable. In contrast to designated blockchains and integrated approaches, ENS provides better security and flexibility. Support for EV certificates and authorities can be added. To prevent past issues with CAs, the ideas described by Wilson and Ateniese [19] can be generalized. Subordinate CAs should be granted the right to transfer funds from the root-level CA to the subscriber. The subscriber would then transfer these back, ensuring that every root-level CA trusts subordinate CAs enough to let them carry out the vetting procedure. Also, certificates should only be trusted in a country in which the CA is liable for misbehavior. Both enhancements ensure more meaningful and secure certificates. It is also possible to support PGP-like trust relationships using a particular resolver. For the full-scale adoption of systems like ENS, there are, however, two major issues that yet have to be overcome.

1) The transaction throughput of current blockchains needs to be increased without economizing on decentralization. Either sharding or 2nd layer solutions could achieve this.
2) Thin client protocols need to be matured and support for suitable cryptographic data structures added to allow for faster and more secure retrieval of identifiers than with SPV.

Both points are a topic of ongoing research and development. We can expect advances in this area as a blockchain's potential is not fully unlocked here. Hence, combined with high security and flexibility, systems like ENS are arguably the future of PKI.

REFERENCES

[1] "Timeline of PKI Security Failures." [Online]. Available: https://sslmate.com/certspotter/failures

[2] R. Sleevi, "Sustaining Digital Certificate Security," 2015. [Online]. Available: https://security.googleblog.com/2015/10/sustaining-digital-certificate-security.html

[3] S. Ruoti, J. Andersen, D. Zappala, and K. Seamons, "Why Johnny Still , Still Can âĂŹ t Encrypt : Evaluating the Usability of a Modern PGP Client," 2016.

[4] J. R. Vacca, *Public Key Infrastructure: Building Trusted Applications and Web Services*, 2004. [Online]. Available: https://books.google.com/books?id=3kS8XDALWWYCpgis=1

[5] N. Alexopoulos, J. Daubert, M. Mühlhäuser, and S. M. Habib, "Beyond the hype: On using blockchains in trust management for authentication," in *Trustcom/BigDataSE/ICESS, 2017 IEEE*. IEEE, 2017, pp. 546–553.

[6] C. Fromknecht and D. Velicanu, "A Decentralized Public Key Infrastructure with Identity Retention," *Cryptology ePrint Archive*, pp. 1–16, 2014.

[7] T. Hepp, M. Sharinghousen, P. Ehret, A. Schoenhals, and B. Gipp, "On-chain vs. off-chain storage for supply- and blockchain integration," *it - Information Technology*, nov 2018. [Online]. Available: https://doi.org/10.1515/itit-2018-0019

[8] M. Cooper, Y. Dzambasow, P. Hesse, S. Joseph, and R. Nicholas, "Internet X.509 Public Key Infrastructure: Certification Path Building," Internet Requests for Comments, RFC Editor, Tech. Rep. 4158, 9 2005. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4158.txt

[9] E. Rescorla, "HTTP Over TLS," Internet Requests for Comments, RFC Editor, Tech. Rep. 2818, 5 2000. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2818.txt

[10] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," Internet Requests for Comments, RFC Editor, Tech. Rep. 5246, 8 2008. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5246.txt

[11] P. Gutmann, "PKI: It's Not Dead, Just Resting," *Computer*, vol. 35, no. 8, pp. 41 – 49, 2002.

[12] "The EFF SSL Observatory." [Online]. Available: https://www.eff.org/observatory

[13] P. R. Zimmermann, *The Official PGP User's Guide*. Cambridge, MA, USA: MIT Press, 1995.

[14] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer, "OpenPGP Message Format," Internet Requests for Comments, RFC Editor, Tech. Rep. 4880, 11 2007. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4880.txt

[15] "Validating other keys on your public keyring." [Online]. Available: https://www.gnupg.org/gph/en/manual/x334.html

[16] G. Caronni, "Walking the Web of trust," *Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE*, vol. 2000-Janua, pp. 153–158, 2000.

[17] A. Barenghi, A. Di Federico, G. Pelosi, and S. Sanfilippo, "Challenging the trustworthiness of pgp: Is the web-of-trust tear-proof?" in *European Symposium on Research in Computer Security*. Springer, 2015, pp. 429–446.

[18] A. Yakubov, W. M. Shbair, A. Wallbom, D. Sanda, and R. State, "A Blockchain-Based PKI Management Framework," 2018.

[19] D. Wilson and G. Ateniese, "From pretty good to great: Enhancing PGP using bitcoin and the blockchain," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9408, pp. 368–375, 2015.

[20] C. Allen, A. Brock, Buterin Vitalik, J. Callas, C. Lundkvist, P. Kravchenko, J. Nelson, D. Reed, and G. Slepak, "Decentralized Public Key Infrastructure," p. 21, 2015. [Online]. Available: http://www.weboftrust.info/downloads/dpki.pdf

[21] D. Atkins and R. Austein, "Threat Analysis of the Domain Name System (DNS)," Internet Requests for Comments, RFC Editor, Tech. Rep. 3833, 8 2004.

[22] "W3C Community Group Draft Report 2018. Decentralized Identifiers (DIDs) v0.11." [Online]. Available: https://w3c-ccg.github.io/did-spec

[23] B. Fredriksson, "A Distributed Public Key Infrastructure for the Web Backed by a Blockchain," 2017.

[24] L. Kolisko, "In-depth on differences between public, private and permissioned blockchains." [Online]. Available: https://medium.com/@lkolisko/in-depth-on-differences-between-public-private-and-permissioned-blockchains-aff762f0ca24

[25] "Ethereum Name Service: Auctions Finalized." [Online]. Available: https://etherscan.io/ens?filter=hashregistered

[26] "Ethereum ENS Registrations Chart." [Online]. Available: https://etherscan.io/chart/ens-register

[27] "Transparency Report: HTTPS encryption on the web." [Online]. Available: https://transparencyreport.google.com/https/certificates

[28] G. Edgecombe, "Google Rocketeer." [Online]. Available: https://ct.grahamedgecombe.com/logs/3

[29] J. Kim, "Vitalik Buterin: Sharding and Plasma to Help Ethereum Reach 1 Million Transactions Per Secondhttps://cryptoslate.com/vitalik-buterin-sharding-and-plasma-to-help-ethereum-reach-1-million-transactions-per-second/." [Online]. Available: https://cryptoslate.com/vitalik-buterin-sharding-and-plasma-to-help-ethereum-reach-1-million-transactions-per-second/

Listing 1: Use the following BibTeX code to cite this article

```
@InProceedings{hepp2019a,
  Title                    = {Exploring Potentials and Challenges of Blockchain
     -based Public Key Infrastructures},
  Author                   = {Hepp, Thomas and Spaeh, Fabian and Schoenhals,
     Alexander and Ehret, Philip and Gipp, Bela},
  Booktitle                = {IEEE INFOCOM 2019 - 2nd Workshop on on
     Cryptocurrencies and Blockchains for Distributed Systems (CryBlock 2019)},
  Address                  = {Paris, France},
  Year                     = {2019},
  Month                    = {May}
}
```