Provably Secure, Smart Contract-based Naming Services: Design, Implementation and Evaluation*

Christos Patsonakis

National and Kapodistrian University of Athens
Department of Informatics and Telecommunications
c.patswnakis@di.uoa.gr

Abstract. Naming services provide the foundations of developing diverse and important applications. Currently, these are operated by centralized authorities, which have to be trusted for their correct operation. Unfortunately, centralization incurs several downsides as illustrated by numerous security incidents where such authorities have been compromised. Decentralization has been proposed as an alternative to deal with these issues. Nevertheless, decentralization raises other issues, such as free-riding and Sybil attacks.
In this thesis, we leverage smart contract platforms and propose the design of a decentralized naming service. We are the first to fully formalize the naming service design problem in the Universal Composability framework and formally prove the security of our construction. The main barrier in realizing a smart contract-based naming service is the size of the contract's state, which should be minimized. We resolve this issue by defining and using in our naming service a public-state cryptographic accumulator with constant size. We propose and implement a second construction, which preserves the security properties of the first and is the only version with constant-sized state that can be deployed on Ethereum's live chain. We compare these constructions with the simple approach of prior works, where all identity records are stored on the contract's state. We address several shortcomings of Ethereum and its cost model by introducing an alternative paradigm for developing smart contract-based applications. Our approach is relevant for a wide range of applications, e.g., any key-value store. We address Ethereum's monotonically increasing state by introducing recurring fees that are adjustable by miners. We propose a scheme where the cost of storage-related operations reflects the effort that miners have to expend to execute them. We show that under such a pricing scheme that encourages economy, the constructions presented in this work reduce incurred transaction fees by up to an order of magnitude.

Keywords: public key infrastructure, smart contract, cryptographic accumulator

* Dissertation Advisor: Mema Roussopoulos, Associate Professor.

1 Introduction

Contrary to its original, clean design principles, the Internet today is not completely decentralized. The Internet's naming services, such as Domain Name Systems (DNSs) and Public Key Infrastructures (PKIs), provide the most critical building blocks that allow and secure, respectively, digital communications. These services handle mappings between entitie names and values (e.g., an IP address for DNSs, or a public key for PKIs). Unfortunately, these critical services are under the control of centralized, remote entities that must be trusted for their correct operation. This is problematic as it is proven by multiple security incidents in e.g. centralized public key infrastructures, where certification authorities have been compromised. (e.g., [64, 72, 83])

The advent of Bitcoin revolutionized the world of digital payments since it was the first system that allowed entities that do not trust each other to transact securely, without relying on trusted, third parties. The operation of Bitcoin is based on a distributed peer-to-peer network that maintains a highly replicated, verifiable, append-only transaction log, which is commonly referred to as a blockchain. This technology is very promising as it allows the development of systems that are completely decentralized. As a result of this potential, there have been calls from the community to re-decentralize the Internet using blockchain technology to build its critical naming services, thereby eliminating the Internet's reliance on centralized entities.

Namecoin ([25]) and Emercoin ([8]) are notable examples of blockchain-based naming services. These systems employ the blockchain to store, query and verify the validity of records pertaining to identities. However, this approach is inefficient for the following reasons. First, it forces client to download and maintain a complete copy of the blockchain to enable them to validate identity records. Second, the computational and storage complexities increase linearly to the number of registered records. Third, the system's usability is severely limited as important devices are excluded due to their limited storage (e.g., smartphones). Finally, it increases the amount of information that network nodes (miners) need to retain, which may prevent new nodes from synchronizing and contributing to the network's security.

In this thesis, we present the design, implementation and evaluation of the first provably secure, fully distributed naming service. The building blocks of our service are smart contracts and cryptographic accumulators. We present two constructions of our service that store constant-sized state on the smart contract, regardless of the number of registered identities. In this way, we fully resolve the aforementioned issues associated with on-blockchain storage. Storage of identity records is offloaded to an external, potentially unreliable, distributed storage network which, among others, allows for more efficient access to identity records.

2 Background

In this chapter, we provide background knowledge that is required to understand the remaining chapters of this thesis. In particular, we present basic concepts about public key cryptosystems, digital certificates and public key infrastructures. Next, we describe the operation of blockchain systems, which form the building block on which the Ethereum smart contract platform was built, on top of which we implemented all the constructions presented in this thesis. Finally, we present an introduction to cryptographic accumulators and their properties.

3 Related Work

In this chapter, we present the results of up to date bibliography on naming services, as well as, public key infrastructures. Based on the fact that several previous approaches use the same underlying primitive for the development of such services, we categorize them based on it. In summary, we review previous works that are based on: 1) replicated state machines, 2) various types of overlay networks, and 3) blockchains. Our bibliographic research allows us to illustrate the weaknesses and failures of each category, as well as, each system individually. We note that no previous approach addresses, overall, issues such as time-stamping, arbitrary failures, free-riding, Sybil attacks, formally proven secure constructions and others.

4 Naming Service: Building Blocks and Definition

In this chapter, we begin by introducing basic notation that is relevant to this thesis. We also formally present definitions of standard cryptographic hardness problems, such as the strong RSA and collision-resistant cryptographic hash functions. Next, we present the first official definition of a public-state, additive, universal cryptographic accumulator, which is the building block of our naming service. Finally, we present the formal definition of our naming service, in the Universal Composability framework, which we model as an ideal functionality. We also present the formal definition of two additional ideal functionalities that model the (unreliable) storage network and the smart contract, respectively. Finally, we provide a general description of how clients interact with all of the above functionalities in our protocols, which we present in the following chapters.

5 RSA-based PKI Construction

In this chapter, we present the first construction of our naming service. We begin by presenting a construction of a public-state, additive, universal cryptographic accumulator whose security is based on the strong RSA assumption in the Random Oracle model. The input domain of this accumulator is limited to prime numbers. However, our naming service requires the accumulation of arbitrary strings. We solve this problem by presenting a polynomial complexity function that associates arbitrary strings to prime numbers. This function is a deterministic version of that of Gennaro et al. [69]. We prove the collision-resistance of this function and that on input an arbitrary string, this function will output a prime number, except with negligible probability. Next, we present the construction of a public key infrastructure that uses the aforementioned building blocks to implement the operations of our naming service. In particular, we present the smart contract's code, as well as, our construction's protocol. Finally, we define the security theorem of this construction, whose proof is provided in the appendix of the thesis.

6 Hash tree-based PKI Construction

In this chapter, we present the second construction of our naming service. We begin by presenting the universal cryptographic accumulator of Camacho et al. [54], which is based on hash trees. This accumulator is *strong*, i.e., it offers more properties than the one we defined in the previous chapter. However, in this accumulator, the structures used to prove whether an element is accumulated or not have are logarithmic, unlike the previous accumulator where they are of constant size. Next, we present the construction of a public key infrastructure that uses the aforementioned accumulator to implement the operations of our naming service. Similar to the previous chapter, we present the smart contract's code, the protocol of our construction, and its security theorem.

7 Evaluation

In this chapter, we present experiments that measure the cost of running on Ethereum the two constructions of our naming service, as well as, their building blocks. We intersperse our results with recommendations for modifications and improvements to Ethereum that, we believe, are vital if Ethereum (or any smart contract platform) is to reach its maximum potential of supporting arbitrary, large-scale, distributed applications.

Our first set of experiments demonstrates the extra cost of implementing a digital signature scheme on a smart contract, compared to the implementation based on precompiled contracts provided by the Ethereum platform. We calculate an average overhead cost of two orders of magnitude.

We then present the evaluation of our two PKI constructions. We present the configuration of each construction (e.g., values for the security parameters) and the cost of each operation of the smart contract. For each construction, we discuss extensively the semantics of the results and illustrate its deployment scale. In summary, for the first construction that is based on the strong RSA assumption, our results show that it cannot be deployed on Ethereum's live chain, due to the high cost of the process that maps arbitrary strings to prime numbers. However, the second construction, which is based on hash trees, can be deploy on Ethereum's live chain for medium-sized PKIs. We complement the results of each construction with a discussion where we propose future optimizations. Finally, we implement and evaluate a PKI where all the records are stored on the smart contract's state, which is the approach followed by all previous blockchain-based naming services. Our results show that, at present, this approach is the most cost-effective. However, the impact of this approach on the future and longevity of the platform is severe. This is a hot topic of discussion in the Ethereum community, which has been looking for ways to address it for years. However, our own constructions are perfectly aligned with the future and longevity of Ethereum, as well as, any other smart contract platform, as they incur constant storage overhead, irrespective of the number of registered identities.

8 An Alternative Paradigm for Developing Applications and Pricing Storage on Smart Contract Platforms

In this chapter, drawing on the research and results of the previous chapter, we present our approach to tackle all issues arising from the practice of using blockchains or smart contracts as a means of directly storing, validating and retrieving data. We introduce an alternative methodology for developing applications on these platforms where we employ the state of smart contracts only to validate application data. Data storage and retrieval are offloaded to an external, possibly unreliable, storage network. Our methodology is generic and can be applied to any key-value store. To support the effectiveness and adaptability of our approach, we present the design and implementation of an ERC20 token standard adaptation. We evaluate our construction and compare it with the standard implementation.

We address the problem of Ethereum's monotonically increasing state by presenting an adaptive model for pricing storage operations. Specifically, we introduce recurring storage fees commensurate with the amount of storage consumed by each smart contract, pricing storage operations based on the complexity of their execution, and freeing up storage space occupied by contracts that are no longer used. Under this cost model, we re-evaluate both the construction of our ERC20 token construction and our hash tree-based PKI construction. Our results show that both constructions provide cost-effective improvements of up to one order of magnitude.

9 Conclusions and Future Work

In this thesis, we presented the design and implementations of a provably secure naming service based on smart contracts. Our service provides a combination of properties that no previous approach has been able to offer. However, interesting technical issues remain to be explored. Indicatively, we mention the following:

– Design, implement, and evaluate of a distributed storage network that is scalable and resistant to Sybil attacks.

– Reduce or eliminate the computational burden that storage network nodes may impose on clients.
– Apply and evaluate verifiable computation techniques in regards to the procedure that maps arbitrary strings to prime numbers.
– Extend the security model of the accumulator of Camacho et al. [54] to reduce the cost of the operations of the hash tree-based construction.

References

1. Amazon elastic file system. https://aws.amazon.com/efs/
2. Ascap, prs and sacem join forces for blockchain copyright system. https://www.musicbusinessworldwide.com/ascap-prs-sacem-join-forces-blockchain-copyright-system/, accessed: 2017-07-06
3. Consensys: Ethereum multisigwallet. https://github.com/ConsenSys/MultiSigWallet
4. Cryptokitties. https://www.cryptokitties.co/
5. Eip 103: Blockchain rent. https://tinyurl.com/yc3uc4ak
6. Eip 1418 blockchain rent: fixed cost per word-block. https://github.com/ethereum/EIPs/issues/1418
7. Eip20 - erc20 token standard. https://tinyurl.com/ycd8mzb3
8. Emercoin - distributed blockchain services for business and personal use. http://www.emercoin.com
9. Erc20 token market capitalization. https://etherscan.io/tokens
10. Erc20 token market capitalization. https://tinyurl.com/yd9fnw9q
11. Erc721 - a class of unique tokens. http://erc721.org/
12. Eth gas station. https://ethgasstation.info/
13. Ether - ethereum homestead 0.1 documentation. https://tinyurl.com/ybcerf39
14. Ethereum - merkle patricia tree. https://tinyurl.com/zl2z4m8
15. Ethereum average gas limit chart. https://tinyurl.com/yaokfvl2
16. Ethereum name service. https://ens.domains/
17. Ethereum price chart us dollar (eth/usd). https://tinyurl.com/jxsjqqd
18. Ethereum wiki - erc20 token standard. https://tinyurl.com/yd9fnw9q
19. Evm opcodes and instruction reference. https://tinyurl.com/ya7o3t6c
20. Github repository of all our implementations. https://github.com/razaden
21. Go ethereum. https://tinyurl.com/jkw5ow9
22. Google: Leveldb. https://github.com/google/leveldb
23. Ibm pushes blockchain into the supply chain. https://www.wsj.com/articles/ibm-pushes-blockchain-into-the-supply-chain-1468528824, accessed: 2017-07-06
24. Ipfs is the distributed web. https://ipfs.io/
25. Namecoin. https://namecoin.org/
26. Nist: Digital signature standard (dss). https://tinyurl.com/ybjakloz
27. Nist: Recommendation for key management. https://tinyurl.com/ybcmxqlv
28. solc-js. https://github.com/ethereum/solc-js
29. Solidity. https://solidity.readthedocs.io/en/v0.5.3/
30. Storj - decentralized cloud object storage that is affordable, easy to use, private, and secure. https://storj.io/
31. Swarm. https://tinyurl.com/y7fz8q3u

32. Swiss industry consortium to use ethereums blockchain. https://www.ccn.com/swiss-industry-consortium-use-ethereums-blockchain/, accessed: 2017-07-06
33. Truffle suite. https://truffleframework.com/
34. Zerocoin: Solidity big number library. https://tinyurl.com/yaa34saq
35. Ethereum?s vitalik buterin wants to create annual ?rent? fees. https://tinyurl.com/yal56med (July 2018)
36. A simple and principled way to compute rent fees. https://tinyurl.com/y9vv6w59 (March 2018)
37. Vitalik wants you to pay to slow ethereum's growth. https://tinyurl.com/y9gj8zvz (March 2018)
38. Aberer, K.: P-grid: A self-organizing access structure for p2p information systems. In: Proceedings of the 9th International Conference on Cooperative Information Systems. Springer-Verlag (2001)
39. Al-Bassam, M.: Scpki: A smart contract-based pki and identity system. In: Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts (2017)
40. Alex Beregszaszi, P.B.: Eip 145. https://tinyurl.com/yc4khbj8
41. Ali, M., Nelson, J., Shea, R., Freedman, M.: Blockstack: A global naming and storage system secured by blockchains. In: USENIX Annual Technical Conferencei (ATC) (2016)
42. androlo: Solidity contracts. https://tinyurl.com/ya9s68dh
43. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., Cocco, S.W., Yellick, J.: Hyperledger fabric: A distributed operating system for permissioned blockchains. In: Proceedings of the 13th EuroSys Conference. ACM (2018)
44. Avramidis, A., Kotzanikolaou, P., Douligeris, C., Burmester, M.: Chord-pki: A distributed trust infrastructure based on p2p networks. Computer Networks **56** (January 2012)
45. Badertscher, C., Gaži, P., Kiayias, A., Russell, A., Zikas, V.: Ouroboros genesis: Composable proof-of-stake blockchains with dynamic availability. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS) (2018)
46. Badertscher, C., Maurer, U., Tschudi, D., Zikas, V.: Bitcoin as a transaction ledger: A composable treatment. In: Proceedings of the 37th Annual International Conference on the Advances in Cryptology (CRYPTO) (2017)
47. Baldimtsi, F., Camenisch, J., Dubovitskaya, M., Lysyanskaya, A., Reyzin, L., Samelin, K., Yakoubov, S.: Accumulators with applications to anonymity-preserving revocation. In: IEEE European Symposium on Security and Privacy (EuroS&P) (2017)
48. Bari, N., Pfitzmann, B.: Collision-free accumulators and fail-stop signature schemes without trees. In: EUROCRYPT (1997)
49. Baylina, J.: Eip 1109. https://tinyurl.com/yckxjogx
50. Benaloh, J.C., de Mare, M.: One-way accumulators: A decentralized alternative to digital sinatures (extended abstract). In: Proceedings of Advances in Cryptology - Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT) (1993)

51. Bonneau, J.: Eth iks: Using ethereum to audit a coniks key transparency log. In: Financial Cryptography and Data Security - International Workshops, FC 2016, BITCOIN, VOTING, and WAHC (2016)
52. Buterin, V.: Eip 198. `https://tinyurl.com/y9mhw6jz`
53. Buterin, V.: Transaction fee economics. `https://tinyurl.com/y8ckvboh`
54. C., P., H., A., K., M.A., O., R.: Strong accumulators from collision-resistant hashing. In: 11th International Conference on Information Security. Springer Berlin Heidelberg (2008)
55. Camenisch, J., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Proceedings of the 22ond Annual Internation Conference on Advances in Cryptology (CRYPTO) (2002)
56. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: IEEE Symposium on Foundations of Computer Science. IEEE Computer Society (2001)
57. Carter, L., Wegman, M.N.: Universal classes of hash functions. Journal of Computer and System Sciences (1979)
58. Choudhury, S., Bhatnagar, K., Haque, W.: Public Key Infrastructure Implementation and Design. John Wiley & Sons, Inc., 1st edn. (2002)
59. Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., Polk, W.: Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile. RFC 5280, RFC Editor (May 2008). https://doi.org/10.17487/RFC5280, `https://www.rfc-editor.org/rfc/rfc5280.txt`
60. Datta, A., Hauswirth, M., Aberer, K.: Beyond "web of trust": Enabling p2p e-commerce. In: IEEE International Conference on E-Commerce (June 2003)
61. Douceur, J.R.: The sybil attack. Revised Papers from the 1st International Workshop on Peer-to-Peer Systems (IPTPS) (2001)
62. Ellison, C., Schneier, B.: Ten risks of pki: What you're not being told about public key infrastructure. Computer Security Journal **16** (December 2000)
63. Feinler, E., Harrenstien, K., Su, Z., White, V.: Dod internet host table specification. RFC 810, RFC Editor (March 1982). https://doi.org/10.17487/RFC0810, `https://www.rfc-editor.org/rfc/rfc810.txt`
64. Fisher, D.: Final report on diginotar hack shows total compromise of ca servers. `https://threatpost.com/final-report-diginotar-hack-shows-total-compromise-ca-servers-103112/77170/`, [Online; posted 31-October-2012]
65. Fromknecht, C., Velicanu, D., Yakoubov, S.: A decentralized public key infrastructure with identity retention. IACR Cryptology ePrint Archive (2014)
66. Garay, J.A., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol: Analysis and applications. In: EUROCRYPT (2015)
67. Garay, J.A., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol with chains of variable difficulty. In: Proceedings of the 37th Annual International Conference on the Advances in Cryptology (CRYPTO) (2017)
68. Garg, N.: Apache Kafka. Packt Publishing (2013)
69. Gennaro, R., Halevi, S., Rabin, T.: Secure hash-and-sign signatures without the random oracle. In: EUROCRYPT (1999)
70. Gipp, B., Meuschke, N., Gernandt, A.: Decentralized trusted timestamping using the crypto currency bitcoin. iConference (2015)
71. GnuPG: Libgcrypt. `https://tinyurl.com/yaa8m7ao`
72. Goodin, D.: Google takes symantec to the woodshed for mis-issuing 30,000 https certs. `https://arstechnica.com/information-technology/2017/03/google-`

takes-symantec-to-the-woodshed-for-mis-issuing-30000-https-certs/, [Online; posted 24-March-2017]
73. Gutmann, P.: Pki: it's not dead, just resting. Computer **35** (August 2002)
74. Housley, R., Ford, W., Polk, W., Solo, D.: Internet x.509 public key infrastructure certificate and crl profile. RFC 2459, RFC Editor (January 1999). https://doi.org/10.17487/RFC2459, https://www.rfc-editor.org/rfc/rfc2459.txt
75. J. Sermersheim, E.: Lightweight directory access protocol (ldap): The protocol. RFC 4511, RFC Editor (June 2006). https://doi.org/10.17487/RFC4511, https://www.rfc-editor.org/rfc/rfc4511.txt
76. Jhanwar, M.P., Safavi-Naini, R.: Compact accumulator using lattices. In: Proceedings of the 5th International Conference on Security, Privacy, and Applied Cryptography Engineering (SPACE) (2015)
77. Karakaya, M., Korpeoglu, I., Ulusoy, O.: Free riding in peer-to-peer networks. IEEE Internet Computing **13** (March 2009)
78. Lamport, L., Shostak, R., Pease, M.: The byzantine generals problem. ACM Trans. Program. Lang. Syst. (1982)
79. Lesueur, F., Me, L., Tong, V.V.T.: An efficient distributed pki for structured p2p networks. In: IEEE P2PC (2009)
80. Li, J., Li, N., Xue, R.: Universal accumulators with efficient nonmembership proofs. In: Proceedings of the 5th International Conference on Applied Cryptography and Network Security (ACNS) (2007)
81. LLC., G.: https://www.chromium.org/Home/chromium-security/root-ca-policy, [Online; posted 04-May-2016]
82. Mashatan, A., Vaudenay, S.: A fully dynamic universal accumulator. Proceedings of the Romanian Academy (2013)
83. Masnick, M.: Trustwave admits it issued a certificate to allow company to run man-in-the-middle attacks. https://www.techdirt.com/articles/20120208/03043317695/trustwave-admits-it-issued-certificate-to-allow-company-to-run-man-in-the-middle-attacks.shtml, [Online; posted 08-February-2012]
84. Melara, M.S., Blankstein, A., Bonneau, J., Felten, E.W., Freedman, M.J.: CONIKS: Bringing key transparency to end users. In: 24th USENIX Security Symposium (USENIX Security) (2015)
85. Mockapetris, P.V.: Domain names - concepts and facilities. RFC 882, RFC Editor (November 1983). https://doi.org/10.17487/RFC0882, https://www.rfc-editor.org/rfc/rfc882.txt
86. Mockapetris, P.V.: Domain names: Implementation specification. RFC 883, RFC Editor (November 1983). https://doi.org/10.17487/RFC0883, https://www.rfc-editor.org/rfc/rfc883.txt
87. Mockapetris, P.V.: Domain names: Implementation and specification. RFC 1035, RFC Editor (November 1987). https://doi.org/10.17487/RFC1035, https://www.rfc-editor.org/rfc/rfc1035.txt
88. Morselli, R., Bhattacharjee, B., Katz, J., Marsh, M.A.: Keychains: A decentralized public-key infrastructure. Technical Reports from UMIACS (2006)
89. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. http://bitcoin.org/bitcoin.pdf, [Online; posted 31-October-2008]
90. Nguyen, L.: Accumulators from bilinear pairings and applications. In: Proceedings of the Conference on Topics in Cryptology - The Cryptographers' Track at the RSA Conference (CT-RSA) (2005)
91. Nyberg, K.: Fast accumulated hashing. In: Proceedings of the 3rd International Workshop on Fast Software Encryption (FSE) (1996)

92. Patsonakis, C., Roussopoulos, M.: An alternative paradigm for developing and pricing storage on smart contract platforms. In: IEEE International Conference on Decentralized Applications and Infrastructures. IEEE (2019)
93. Patsonakis, C., Samari, K., Roussopoulos, M., Kiayias, A.: Towards a smart contract-based, decentralized, public-key infrastructure. In: 16th International Conference on Cryptology and Network Security. Springer International Publishing (2018)
94. Patsonakis, C., Samari, K., Roussopoulos, M., Kiayias, A.: On the practicality of a smart contract pki. In: IEEE International Conference on Decentralized Applications and Infrastructures. IEEE (2019)
95. Raju, P., Kadekodi, R., Chidambaram, V., Abraham, I.: Pebblesdb: Building key-value stores using fragmented log-structured merge trees. In: Proceedings of the 26th Symposium on Operating Systems Principles (SOSP) (2017)
96. Raju, P., Ponnapalli, S., Kaminsky, E., Oved, G., Keener, Z., Chidambaram, V., Abraham, I.: mlsm: Making authenticated storage faster in ethereum. In: 10th USENIX Workshop on Hot Topics in Storage and File Systems, HotStorage (2018)
97. Reiter, M.K., Franklin, M.K., Lacy, J.B., Wright, R.N.: The omega key management service. In: ACM Conference on Computer and Communications Security (CCS) (1996)
98. Rosser, B.: Explicit bounds for some functions of prime numbers. American Journal of Mathematics (1941)
99. Salowey, J., Turner, S.: Iana registry updates for tls and dtls. RFC 8447, RFC Editor (August 2018). https://doi.org/10.17487/RFC8447, `https://www.rfc-editor.org/rfc/rfc8447.txt`
100. Slagell, A., Bonilla, R., Yurcik, W.: A survey of pki components and scalability issues. In: IEEE International Performance Computing and Communications Conference (April 2006)
101. Sousa, J., Bessani, A., Vukolic, M.: A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. In: Proceeding of the 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) (2018)
102. Szabo, N.: Formalizing and securing relationships on public networks. First Monday **2** (1997)
103. Ta-Shma, T.S.A., Yung, M.: Blind, auditable membership proofs. In: Proceedings of the 4th International Conference on Financial Cryptography (FC) (2000)
104. Tomescu, A., Devadas, S.: Catena: Efficient non-equivocation via bitcoin. In: IEEE Symposium on Security and Privacy (SP) (2017)
105. Turner, S.: The application/pkcs10 media type. RFC 5967, RFC Editor (August 2010). https://doi.org/10.17487/RFC5967, `https://www.rfc-editor.org/rfc/rfc5967.txt`
106. V. Buterin: A simple and principled way to compute rent fees. `https://tinyurl.com/y9vv6w59`
107. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. `https://github.com/ethereum/yellowpaper`, [Online; posted April-2014]
108. Wood, G.: Ethereum yellow paper. `https://tinyurl.com/yaptyawg`
109. Wouhaybi, R.H., Campbell, A.T.: Keypeer: A scalable, resilient distributed public-key system using chord (2008)
110. Wu, X., Xu, Y., Shao, Z., Jiang, S.: Lsm-trie: An lsm-tree-based ultra-large key-value store for small data items. In: USENIX Annual Technical Conference (USENIX ATC) (2015)

111. Y. Dong, W.K., Boutaba, R.: Conifer: centrally-managed pki with blockchain-rooted trust. In: IEEE International Conference on Blockchain (Blockchain) (2018)
112. Yakubov, A., Shbair, W.M., Wallbom, A., Sanda, D., State, R.: A blockchain-based pki management framework. In: IEEE/IFIP Network Operations and Management Symposium (NOMS) (2018)
113. Yüce, E., Selçuk, A.A.: Server notaries: a complementary approach to the web pki trust model. IET Information Security **12** (September 2018)
114. Zhou, L., Schneider, F.B., Van Renesse, R.: Coca: A secure distributed online certification authority. ACM Transactions on Computer Systems (TOCS) (2002)
115. Zimmermann, P.R.: The Official PGP User's Guide. MIT Press (1995)