

AuthLedger: A Novel Blockchain-based Domain Name Authentication Scheme

Zhi Guan^{1,3}, Abba Garba^{2,3}, Anran Li^{1,3}, Zhong Chen^{1,3} and Nesrine Kaaniche^{2,3}

¹*Institute of Software, EECS, Peking University, National Engineering Research Center for Software Engineering, Peking University, Beijing, China*

²*MoE Key Lab of High Confidence Software Technologies, Peking University, Beijing, China*

³*SAMOVAR, Telecom SudParis, CNRS, University of Paris-Saclay, France*

Keywords: PKI, Blockchain, Authentication, Cryptography.

Abstract: Nowadays public key infrastructure authentication mainly rely on certificate authorities and have to be trusted by both domain operators and domain owners. Domain Name System Security Extensions (DNSSEC) using DNS-based Authentication Name Entities (DANE) DNS records types, offer additional security for authenticating data and integrity to domain name system (DNS). This method allow client via signed statements to specify which CAs are authorized to represent certificate of a domain. Another method is Certificate Authority Authorizations (CAA) developed by Internet Engineering Task Force (IETF) to provide security guarantee against rogue certificate authorities that offer fake certificate for the domain. However, all of these approaches are prone to single point of failure due to their trust attached to infrastructure like Internet Corporation for Assigned Names and Numbers (ICANN). In order to weaken the level of trust to the CAs over certificates, it is necessary to balance the distribution rights among the entities and improve the control of certificate issuance for the certificate owners. Recently with the emergence of Blockchain, a public and distributed ledger, several applications appeared taking advantage of this powerful technology. In this paper, we present an *AuthLedger* a domain authentication scheme based on blockchain technology. The proposed scheme is multi-fold. First, we proposed a domain authentication scheme to reduce the level of trust in CAs. second, we implement our system using Ethereum smart contract. Third, we evaluate security and performance of the proposed system.


1 INTRODUCTION


In current Public Key Infrastructure (PKI) systems, there is a disparity of rights between the different involved entities such as users and certificate authorities (Scheitle et al., 2018). Indeed, users may initiate a certificate signing request, but are not allowed to judge whether the issuing authority has the right to issue certificate for the domain. In fact several breaches of reporting certificates mis-issue have been revealed (Kamat and Gautam, 2018). For instance in September 2015, Google announced to stop using Symantec's Extended Validation certificates due to Symantec issued a certificates without authoriza-


tions for google domains (Rashid, 2015). Meanwhile, in 2015 Lenovo Superfish installed a local CA in its products. This CA is used to inject ads into the transport layer security (TLS) protected web sites to steal confidential data (Kamat and Gautam, 2018). Since the CA private keys are in the computer RAM, they may be easily used to self-examine the traffic. In may, 2016 Symantec bought Blue coat to become part of its subCA with intention to enhances security in the cyber environment. Blue coat has a devices to pry an encrypted internet traffic (Kamat and Gautam, 2018).


These lethal phenomena lead to many researches to disseminate the absolute trust on CAs to various authorities (Khan et al., 2018). Nowadays, certificate miss-issuance is hard to detect due to lack of standard mechanisms to check which certificate authorities are allowed to issue certificates for the domains (Kubilya et al., 2018).


Recently Domain Name System Security Extensions (DNSSEC) offer additional security for au-

^a <https://orcid.org/0000-1111-2222-3333>

^b <https://orcid.org/1111-2222-3333-4444>

^c <https://orcid.org/2222-3333-4444-5555>

^d <https://orcid.org/3333-4444-5555-6666>

^e <https://orcid.org/4444-5555-6666-7777>

thenticating data for domain name system (DNS). DNSSEC records allowed client via signed statements to specify which CAs are authorized to represent certificate of the domain (Gourley and Tewari, 2018). Certificate authority authorizations (CAA) is developed by Internet Engineering Task Force (IETF) as explained in [RFC 6844]¹ to provides security guarantee against rogue CAs. In CAA, domain owners decides which CAs can issue a certificate for their domain via CAA *resource records* (Karaarslan and Adiguzel, 2018). Baldi et al., (2017) describe the DNSSEC system require a lots of complex security requirements for deployment. From one hand, Certificate authority authorization (CAA) has not been deployed by vast majority of the CAs (Hari and Lakshman, 2016). Consequently, latter approaches are more prone to central of failure due to their trust attached to infrastructure like Internet Corporation for Assigned Names and Numbers (ICANN)²(Berkowsky and Hayajneh, 2017).

Blockchain is a decentralized global ledger that contain a series of transactions in the form of blocks (Ali et al., 2016). Each block is secured by hash function to link to another Block in an orderly manner to form a Blockchain network (Yakubov et al., 2018). Ethereum is the second largest blockchain system in terms of value. The objective of such system is to store an arbitrary state in a distributed temper-proof manner (Matsumoto and Reischuk, 2016). Unlike Bitcoin, Ethereum used Turing-complete language and *EthereumVirtualMachine (EVM)* to represent language and computations.

In this paper, we propose *AuthLedger* a novel domain authentication scheme based on blockchain technology. The proposed mechanism records a set of trusted CA associated with each specific domain on the blockchain. That is, each CA has to first verify if is considered to be trusted in order to perform issuance process. On the other hand, each domain can update its trusted CA's on the blockchain and check the issued certificate related to it.

Thus, our contributions are summarized as follows:

- We propose a new solution based on the blockchain technology for identity authentication without a trusted third party.
- Provides an efficient and trustworthy certificate authentication process.
- We implement and conduct an experimental performances' analysis to validate the proposed sys-

tem using Ethereum solidity smart contract environment.

- Analyze security implication of the proposed scheme by discussing different security threats and countermeasures.

The remainder of this paper is organized as follows. Section 2 reviews PKI systems, highlights related security challenges and gives a general overview of the blockchain technology. We present proposed system in 3; 4 we give a detail evaluation of the proposed system, security and performance including the browser extensions; Finally conclusion and future work in section 5.

2 BACKGROUND AND RELATED STUDY

2.1 Certificate Authorities (CAs)

Certificate Authorities involve several hardware and software entities to manage the PKI system (Aishwarya et al., 2015). Certificate based authentication used X.509 PKI standard to provide a strong level of clients authentication. Authentication occur using a third party certification authorities (CA)(Kiayias et al., 2017). Below describe the entities involve in the Certificate of authorities: A *client* will provide information to a CA that verify user identity. *Registration authority (RA)* managed CA task such as authenticating users. *Validation authority (VA)* verify and confirm whether digital certificate is used by adequate trustworthy CA. Different entities involved in the CA as shown in figure 1:

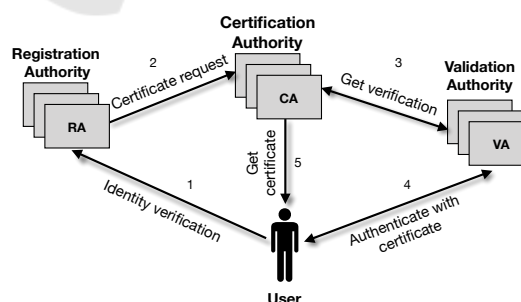


Figure 1: Entities involves in CA.

2.2 DNS Security Extensions (DNSSEC)

DNS security extensions (DNSSEC) using DNS based authentication of named entities (DANE) was designed based on Internet Engineering Task

¹<https://tools.ietf.org/html/rfc6844>

²<https://www.icann.org/>

Force (IETF) [RFC6394-RFC6698] standard to protect cache poisoning attacks (Shulman and Waidner, 2017). The main purpose of DNS security extensions (DNSSEC) is to address some security challenges of the existing PKI certificates authentication (Gourley and Tewari, 2018). However, Security of DNSSEC inherently lies on PKI root at ICANN³ which is vulnerable to central failure. Consequently, vulnerability is much easier to exploit with current DNSSEC system (Sehgal and Dixit, 2019). The Key corrective measure is to ensure domain holders and domain operators behave honestly (Matsumoto et al., 2017). Despite the current proposed solutions there is no standard method to incentivize an entities that behave honestly (Matsumoto and Reischuk, 2016).

Furthermore, DNSSEC using DANE allows domain operators to make judgment about CA based on the statement related to each PKI certificate (Qin et al., 2017). it includes the following information:

- **CA Restrictions:** the client should determine which CA should issue certificates.
- **Service Certificate Restrictions:** the client should accept only certain certificates.
- **Trust Anchor:** the client should validate certificates for a particular domain based on the set of available domains provided chain of trust.

2.3 Certification Authority Authorization(CAA)

Certificate authority authorization provides an additional measures to protect issuing accidental certificates (Scheitle et al., 2018). The success of the Certification Authority Authorization (CAA) as a largest DNS security mechanism relies on smooth cooperation between CAs, DNS operators and domain owners. Generally, DNS based extension Aishwarya et al.,(2015); Certificate authority authorization Scheitle et al., (2018); are DNS resource record types built to assist in certificate issuance and verification.

2.4 Blockchain based PKI

Blockchain technology allow an execution of arbitrary logic known as programmable contract. Smart contract is a program that executes on the blockchain by network of mutually distrustful nodes without requiring a trusted authority (Wang et al., 2018). Several solutions are proposed based on the blockchain technology to address the current PKI challenges namely: Namecoin (Kalodner et al., 2015) proposed

a namespace blockchain based system that provide a noble solution to the current challenges of decentralized namespaces. Ali et al. (2016) BlockStack is an implementation of a new type of decentralized internet infrastructure focusing on decentralized application layer of traditional internet architecture.

3 NOVEL BLOCKCHAIN BASED DOMAIN NAME AUTHENTICATION SCHEME

3.1 Threat Model

First we introduce the threat model. In particular we describe several adversaries capabilities: From Blockchain, to malicious client entities that may launch a colluding attacks (between several compromised CAs).

3.1.1 Adversary Capabilities

First scenario we assume there are M full nodes in the blockchain network, in which the entire nodes with p proportion is controlled by malicious attackers, and the other nodes of $1 - p$ are honest full nodes; the number of verification nodes is N , among which q accounts for the verification node is controlled by a malicious attacker, and the other $1 - q$ proportion of the verification nodes are honest nodes.

Additionally we assume malicious entity replicate public keys as the authenticating node to launch (*Sybil attack*).

Second scenario, from *client server* side we assume CAs and validating authorities are malicious which act arbitrarily such as binding fake certificate. Moreover, Domain name server (*DNS*) is assume to be corrupted. We assume that malicious entities cannot collude the hash function of the standard cryptographic protocols.

We also assume when *Browser* is performing checking relevant to a CA may filter erroneous certificates.

3.2 Entities and Its Functionalities

Our proposed system consist of five entities namely:

- **Certificate Authority (CA).** It represents each authorized entity to be able to issue digital certificates in this case, to join AuthLedger to register the information on the Blockchain.

³<https://www.icann.org/>

- **Domain Name Server (DNS).** Maintains the directory of the certificate owner and identity binding.
- **Browser Extension.** Complete domain name Transport layer security (TLS) set up.
- **Validating Authority.** Put vigilance during CA operations for suspicious certificates and report any misbehaviour for an entity.
- **Blockchain.** Which contain full nodes verify binding request and confirm request from validating authorities.

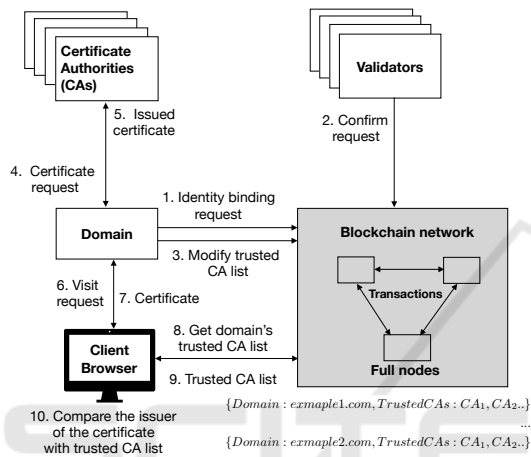


Figure 2: Entities and its Functionalities in AuthLedger.

As depicted in Figure 2 [1] Domain initiates an identity binding request in the blockchain.[2] Validating authority via full nodes verify the requests. [3] A domain updated a trusted CAs in the Blockchain. [4-5] The domain name sends a CA list that is trusted by blockchain network to complete certificate issuance process. [6-7] Client initiates a secure connection through the browser-plug-in to obtain the certificate own by the domain name. [8-10] Browser then requests trusted CAs list from blockchain and compare the validity of the information obtained.

3.3 Domain Authentication

In order to achieve the entity authentication in this paper we provide an authentication procedures based on time [T] and count [C], detail process describe below:

The identity of the binding process is broadly divided into 3 steps:

1. The domain name send an authentication request to the blockchain.

2. After transaction confirmation, the transaction and block are placed on the server side for verification.
3. Node is verify the transaction after the verification period or number of times takes to complete the identity binding.

3.3.1 Authentication Verification based on Time

In time-based authentication process when the verification reaches a specified time it owns the domain name server and completes the binding procedures. As depicted in figure 3.

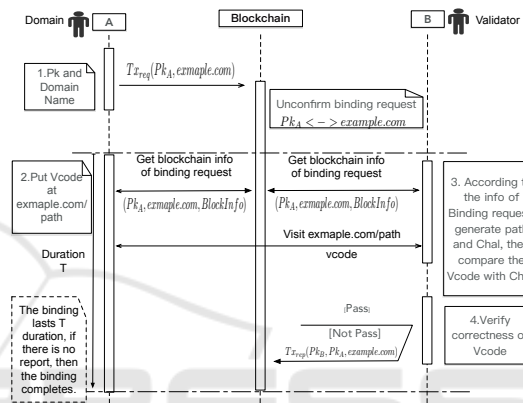


Figure 3: Time Based Domain Authentication.

Time based Domain Process: Hypothesis

Suppose A owns key pairs and domain name e.g $example.com$ such that: $(Pk_A || Sk_A || Dm_{example.com})$; verifier B with the following parameters of key pairs: $(Pk_B || Sk_B)$

Transaction Type

In this scenario, the following three types of transactions are:

- **Binding Transaction Request:** $T_{xrep}(Pk_A || Dm_{example.com})$ binder issues a binding request containing its own public key Pk_A and domain name $example.com$.
- **Report Transaction:** T_{xrep} while processing the binding, the verification node finds that the binder's authentication information is incorrect. Transaction can be reported and rejected.
- **Verify Transaction:** T_{xrep} reporting transaction can be verified by the validator.

Binding Procedures. The specific process of binding is shown in Figure 3, which includes the following steps:

- A Publishes the binding transaction $T_{xreq}(Pk_A, example.com)$ to the blockchain.

- Validator B accesses the domain name to verify that A operates correctly. If not, sends a report transaction $T_{xrep}(Pk_B, Pk_A, example.com)$ to the blockchain.
- After A maintains the verification time T , the verification service can be stopped and the binding is completed.

3.3.2 Authentication Verification based on Count

In count based authentication system verification process reaches the specified number of counts to complete the identity binding as shown in figure 4.

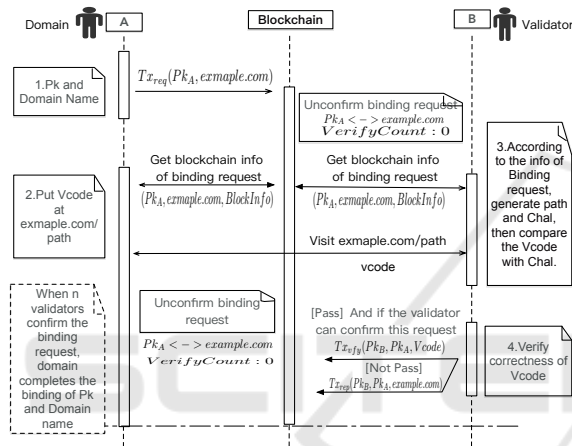


Figure 4: Count Based Domain Authentication Framework.

Count-based Domain Process: Hypothesis. Suppose A owns key pairs and domain name $example.com$ such that: $(Pk_A || Sk_A || Dm_{example.com})$; verifier B owns key pairs: $(Pk_B || Sk_B)$.

Transaction Type. In count based the transaction verification process includes the followings:

- **Binding Transaction Request:** $(Pk_A || Dm_{example.com})$ binder issues a binding request containing its own public key and domain name $example.com$.
- **Reporting Transaction:** T_{xrep} if the verification node finds that the binding information is not accurately placed then identity binding can be rejected.
- **Verification Transaction:** T_{xvy} validator completes the comparison of the verification information, the validator sends the transaction that has passed the verification process.

Binding Process

Authentication process based on the number of validations require more interaction between the validators and the blockchain than time based.

- A Publishes binding transaction $T_{xreq}(Pk_A, ||example.com)$ on the blockchain.
- A gets the block information $Info_{block}$ of the location where T_{xrep} is located, and calculates its value $(Path, Chal) = F(Info_{rcv}Block)$ to put it under domain control.
- B obtains the verification content from $example.com/Path$, submits the transaction $T_{xvy}(Pk_B, Pk_A, Vcode)$ to complete the certificate verification. Once the K has been verified, the binding is complete. The entire process same as time-based requiring the binder to publish her public key Pk_A , domain name $example.com$ onto the blockchain.

3.4 Misbehaviour Incentives

In order to encourage enough verification nodes to join the system, rewards should be given to the verification nodes. As shown in table 1:

First Rule In table 1, fee is deposited by the initiator, instead of unlimited arbitrary identity binding operations; thus avoiding malicious adversary to initiate unlimited invalid identity binding. **Second rule** prevent the verification node from randomly initiating a *reporter* transaction to disrupt the completeness of the normal binding; **The third rule** node that incentivises the verification operation to attract more nodes, but needs to deposit a fee to prevent some nodes from misbehaving **Fourth rule** same as third rule attract more nodes to join the system.

Property 3: Denial Binding Analysis. We assume when binding authentication request is initiated, identity binding is completed after waiting for the validator to complete the verification. However, in case malicious attacker may disrupt the binding. The probability of successfully disrupting this request is shown below:

$$Pr_{denial} = \prod_i = 0^{n/2} \frac{qN - i}{N - i} \quad (1)$$

Property 4: Sybil Attack. Blockchain is built based on open P2P network any node can generate number of identities to increase the probability as verification node. Therefore, additional information is required including a verifiable workload certificate.

Security Comparison of the PKI Authentication system as describe in table 2.

Table 1: Table describe incentive parameters.

Txt. Name	Initiator	PK	Txt. Def.	Txt. Fee	Incentives	Balance
T_{xrep}	Domain Name	Pk_A	Binding Identity	$-P_1$	0	Balance (Pk_A) $-P_1$
T_{xrep}	Reporter	Pk_C	Reporting	$-P_2$	$R_1 = P_2 + \frac{P_1}{2}$	Balance (Pk_C) $+ \frac{P_1}{2}$
$T_{X_{vfy1}}$	Validator	Pk_B	Validation	$-P_3$	$R_2 = P_3 + \frac{P_1}{K}$	Balance (Pk_B) $+ \frac{P_1}{K}$
$T_{X_{vfy2}}$	Validator	Pk_B	Confirm	$-P_3$	$R_3 = P_3 + \frac{P_1}{2n}$	Balance (Pk_B) $+ \frac{P_1}{2n}$

Table 2: Table describe security comparison of the proposed system.

	DNSSEC-DANE	CAA	AuthLedger
Level of Trust	Weak	Weak	Strong
50% attack protection	N/A	N/A	Strong
Sybil attack protection	N/A	N/A	Strong
Fake binding detection	Weak	Weak	Strong
Domain compromise	Strong	Strong	weak
Misbehaviour Incentives	Weak	Weak	Strong
Browser validation	Weak	Weak	Strong

Domain Server Perspective. includes the following properties:

Property 1: We assume most of the Certificate authorities and validating authorities are malicious which act arbitrarily such as binding fake certificate or certificate can be issued from invalid CA. We assume that Domain name server (DNS) are corrupted.

Property 2: Browser client. Additional checks need to be conducted from the client side to make sure any certificate issued by adversary is detected. Therefore browser need to interact with Blockchain to query whether the received certificate is granted by a trusted CA.

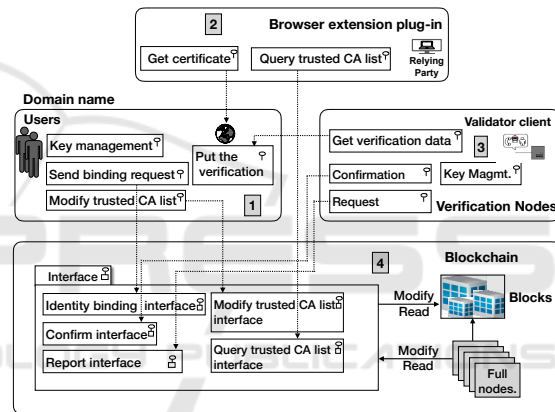


Figure 5: System Architecture.

4 EVALUATION

4.1 Implementation Prototype

Entities involve in the implementation process include the followings: **1. User:** domain sends identity binding request and update trusted CA list to interact interface in the Blockchain. **2. Relying Party:** relying party obtains a certificate through a browser and establishes secure communication, it needs to check the validity of the certificate through Blockchain by query trusted CA list. **3. Verification Nodes:** verify any domain name initiated authentication requests on the blockchain. Also verification node needs to monitor and report any misbehaviour during verification process. **4. Full node:** Monitor all transactions in the Blockchain network. Each time a block is created, the initiator will get the reward corresponding to the proportion of the computational power use. Figure 5

describe the system architecture of the proposed system.

4.2 Smart Contract Solidity

All entities in the proposed system need to be connected through a blockchain. The functions in the smart contract include the followings:

The domain name authentication function: When the domain name calls this function, a series of verification nodes are selected to complete the verification through the verification node.

Trusted list storage: Update the data stored in the smart contract and complete the control of its own trusted CA.

Trust list query: complete the query of the trusted CA list of a specific domain name.

Implementation of these modules relies on Ethereum smart contract to perform the functions of the above interfaces using Solidity programming

Table 3: Trade execution cost.

Name of Trade	Send Data	Tx. Fee (Gas)	Impl. Cost (Gas)	TTL Cost (Gas)	Price (\$)
Binding request	$Pk_A, example.com$	189451	165555	355006	1.977
Verification operation	$Pk_A, example.com, V_{code}$	208475	197342	405 817	2.26
Modify trust list operation	"CA List"	36906	14674	51580	0.287

language. Moreover, Solidity is used to complete the mapping between the Ethereum contract address and the domain name. Sample code for interactions describe below.

```

struct Domain {
    string name;
    uint count;
    string trustCAs;
    uint stBlock;
    address[] validator;
    address addr;
    bool isEntry;
}
mapping(address => Domain) reg\_domain;
mapping(string => Domain) auth\_domain;
uint constant auth\_times = 10;
uint constant limit\_blocks = 100;

```

Meanwhile based on the the underlying blockchain platform, transactions that cause any changes in the Ethereum blockchain need to consume so-called **Gas**. In order to call transaction on ethereum smart contract, the transaction needs to be sent to the blockchain first. The required cost is called *transaction costs*, which is calculated according to the large size of the data. While *execution cost*, it needs to be based on the calculation of the transaction to perform cost assessment. The current Gas station estimate is : $1\ gas = 3\ Gwei^4$. The cost of the binding and trust list modification for *example.com* describe below:

4.3 Browser Plug-in Validation

Browser Selection. The main operation of plug-in is to obtain the trust CA list of the domain name in the blockchain and and compares with the certificate issuer.

4.4 Experiment

In order to test the performance of the proposed system, we used Ali Cloud server configuration is as fol-

⁴<https://ethgasstation.info/>

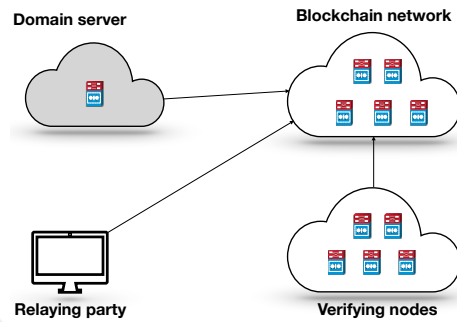


Figure 6: System configuration.

lows: CPU: 1 core, Operating System: Ubuntu 16.04 (64 bit), Memory: 2GB, Disk: 40GB, Golang: 1.8. and Ethereum: Titanium (v1.8.7). Meanwhile machine configuration we used, CPU: 8 cores, Operating System: macOS High Sierra (version 10.13.3) and Chrome: Version 66.0.3359.139 (64-bit).

Blockchain network contain five Ali cloud servers: full node; to complete the blockchain maintenance network; the other five servers as the authentication nodes, each server starts 10 authentication clients, a total of 50 authentication nodes; one Ali cloud deploys web services as a domain name server and runs the domain name guest. The local PC acts as a relaying party, simulates the access to the domain name and completes the acquisition of the trusted CA list.

5 CONCLUSION AND FUTURE WORK

In this paper, we proposed AuthLedger a novel Blockchain-based domain name authentication scheme using Ethereum smart contract to allows a client to trust which CA can issue a certificate for the domain using Blockchain technology. The paper demonstrated an efficient and trustworthy algorithm for certificate authentication process. we also analyze security implication of the proposed scheme by discussing different security threats and countermea-

tures. Moreover, since this is a short paper, future work will concentrate on detail implementation of the AuthLedger to get more experimental results. Will conduct a detailed performance and usability analysis of the proposed system.

REFERENCES

- Aishwarya, C., Raghuram, M., Hosmani, S., Sannidhan, M., Rajendran, B., Chandrasekaran, K., and Bindhumadhava, B. (2015). Dane: An inbuilt security extension. In *Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on*, pages 1571–1576. IEEE.
- Ali, M., Nelson, J. C., 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.
- Berkowsky, J. A. and Hayajneh, T. (2017). Security issues with certificate authorities. In *Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), 2017 IEEE 8th Annual*, pages 449–455. IEEE.
- Gourley, S. and Tewari, H. (2018). Blockchain backed dnssec. In *1st Workshop on Blockchain and Smart Contract Technologies - 21st International Conference on Business Information Systems, Fraunhofer FOKUS, Berlin, 18-20 July, 2018*, pages 357–388. Fraunhofer FOKUS, Berlin.
- Hari, A. and Lakshman, T. (2016). The internet blockchain: A distributed, tamper-resistant transaction framework for the internet. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, pages 204–210. ACM.
- Kalodner, H. A., Carlsten, M., Ellenbogen, P., Bonneau, J., and Narayanan, A. (2015). An empirical study of namecoin and lessons for decentralized namespace design. In *WEIS*. Citeseer.
- Kamat, P. and Gautam, A. S. (2018). Recent trends in the era of cybercrime and the measures to control them. In *Handbook of e-Business Security*, pages 243–258. Auerbach Publications.
- Karaarslan, E. and Adiguzel, E. (2018). Blockchain based dns and pki solutions. *IEEE Communications Standards Magazine*, 2(3):52–57.
- Khan, S., Zhang, Z., Zhu, L., Li, M., Safi, K., Gul, Q., and Chen, X. (2018). Accountable and transparent tls certificate management: An alternate public-key infrastructure with verifiable trusted parties. *Security and Communication Networks*, 2018.
- Kiayias, A., Russell, A., David, B., and Oliynykov, R. (2017). Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*, pages 357–388. Springer.
- Kubilay, M. Y., Kiraz, M. S., and Mantar, H. A. (2018). Certledger: A new pki model with certificate transparency based on blockchain. *arXiv preprint arXiv:1806.03914*.
- Matsumoto, S. and Reischuk, R. M. (2016). Ikp: Turning a pki around with blockchains. *IACR Cryptology ePrint Archive*, 2016:1018.
- Matsumoto, S., Reischuk, R. M., Szalachowski, P., Kim, T. H.-J., and Perrig, A. (2017). Authentication challenges in a global environment. *ACM Transactions on Privacy and Security (TOPS)*, 20(1):1.
- Qin, B., Huang, J., Wang, Q., Luo, X., Liang, B., and Shi, W. (2017). Cecoin: A decentralized pki mitigating mitm attacks. *Future Generation Computer Systems*.
- Rashid, F. Y. (2015). *Google threatens action against Symantec-issued certificates following botched investigation*.
- Scheitle, Q., Chung, T., Hiller, J., Gasser, O., Naab, J., van Rijswijk-Deij, R., Hohlfeld, O., Holz, R., Choffnes, D., Mislove, A., et al. (2018). A first look at certification authority authorization (caa). *ACM SIGCOMM Computer Communication Review*, 48(2):10–23.
- Sehgal, A. and Dixit, A. (2019). Securing web access—dns threats and remedies. In *Emerging Trends in Expert Applications and Security*, pages 337–345. Springer.
- Shulman, H. and Waidner, M. (2017). One key to sign them all considered vulnerable: Evaluation of dnssec in the internet. In *NSDI*, pages 131–144.
- Wang, Z., Lin, J., Cai, Q., Wang, Q., Jing, J., and Zha, D. (2018). Blockchain-based certificate transparency and revocation transparency. *Financial Cryptography and Data Security*. Springer International Publishing.
- Yakubov, A., Shbair, W., Wallbom, A., Sanda, D., et al. (2018). 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 2018, Taipei, Taiwan 23-27 April 2018*.