Implementation of Blockchain Domain Control Verification (B-DCV)

David Khoury*, Patrick Balian*, and Elie Kfoury†
*American University of Science and Technology, Achrafiye, Lebanon
†University of South Carolina, Columbia, United States

***Abstract*—Security in the communication systems rely mainly on a trusted Public Key Infrastructure (PKI) and Certificate Authorities (CAs). Besides the lack of automation, the complexity and the cost of assigning a signed certificate to a device, several allegations against CAs have been discovered, which has created trust issues in adopting this standard model for secure systems. The automation of the servers certificate assignment was achieved by the Automated Certificate Management Environment (ACME) method, but without confirming the trust of assigned certificate. This paper presents a complete tested and implemented solution to solve the trust of the Certificates provided to the servers by using the blockchain platform for certificate validation. The Blockchain network provides an immutable data store, holding the public keys of all domain names, while resolving the trust concerns by applying an automated Blockchain-based Domain Control Validation (B-DCV) for the server and client server verification. The evaluation was performed on the Ethereum Rinkeby testnet adopting the Proof of Authority (PoA) consensus algorithm which is an improved version of Proof of Stake (PoS) applied on Ethereum 2.0 providing superior performance compared to Ethereum 1.0.**

***Keywords*—Ethereum, Blockchain, Public keys, PKI/CA, ACME, SSL/TLS, Trust, Provable, Security, Authentication, DCV;**

I. Introduction

The security and authentication in major components of IT networks rely on public key certificates provided by a trusted third-party (PKI/CA). Lately, the automation of certificate generation was achieved by the Automated Certificate Management Environment (ACME) method from Let's Encrypt [1]. It has solved the automation of the certificate's configuration but without any indication that the certificate is trusted which continues to be a potential issue. The hijacker can use the DNS or BGP to redirect the domain session validation to another server and get a false certificate. When the domain is once validated, there is a possibility to disable it which is the same scenario as with HPKP [2]. The attackers can downtime the sites or can control temporarily the sites and modify the key to the one they have and control. Multiples measures have been investigated and implemented to protect against the interception of the domain validation attacks but there were not covering all cases of validation attacks [3],[4],[5]. The CAs centralization which considered as single point of failure is the main reason for trust issues, as the CA server could be vulnerable to security attacks like the Denial of Service (DoS).

The solution described in this paper proposes a method to resolve the trust concerns of the PKI/CA infrastructure by using the blockchain platform as another trusted platform for certificate validation. The problem of having compelled certificates attack (where a government can compel a CA to issue a valid certificate that can be used for surveillance [6]) will be implicitly resolved, as the CA is not the only party doing the verification, and hence governments or other organizations cannot interfere the verification process.

The paper presents an implementation and testing on Ethereum 2.0 of the Blockchain-based Domain Control Validation (B-DCV) at the server and client side. This work is considered as follow up of the previous paper[7] where a high-level solution was sketched. The novel that a complete test bed for the proposed solution was set up where the client and the smart contract software were developed. The test and the evaluation of the solution was performed on the Rinkeby testnet applying the Ethereum 2.0 based on a Proof of Authority (PoA) consensus algorithm. The results were compared to the existing methods of creating certificates. The output of this work could be considered a potential standardised protocol and applied as an extension to the SSL/TLS client-server secure sessions establishment. The first part of this paper describes the implementation and testing of proposed advanced method of validating server's CA based on Ethereum blockchain network, called "Blockchain-based Domain Control Validation" (B-DCV). This method resolves the trust concerns of assigning certificates for the servers using the currently verification method used by the CAs which is the Domain Control, Validation (DCV), or the new automated method ACME. The B-DCV method does the domain verification process without the need of a trusted certificates. Moreover the B-DCV resolves DDoS attacks on the CAs. The second part describes a new method to establish a SSL/TLS protocol secure session between a client and a server based on the B-DCV method to verify the public key of the server. We demonstrate the feasibility of this method between a web browser and a webserver.

The paper primary contributions consist of: (1) Validating the server's certificate trust concerns based on an automated method called (B-DCV) based on Blockchain,(2)Removing the requirements and the need of domain verification process of a trusted CAs, (3)Eliminating the DDoS attacks targeting a trusted CAs, (4) Developing a software for secure client server session establishment, as a plug-in integrated in browsers

along with a verification server.(5) Adding the ability for browser plugins to verify Name-to-Address resolution of the DNS server when client visits a website. The rest of this paper is divided as follows. A background on generating a certificate and blockchain is provided in Section II. We elaborate and explain in Section III the proposed system and the details of its architecture, components, and configuration. In Section IV, we describe the testing set up and analyze the results outputs. Section V we compare the proposal result with the existing methods of certificate assignment, Section VI summarize the conclusion remarks and discusses the potential future direction of the proposal.

II. BACKGROUND AND RELATED WORK

The installation of a Certificate in a webserver requires the server to use encryption key algorithm software to generate the server's public key and interface a trusted PKI/CA to assign a public key certificate to the server. The ownership verification of the server identity is critical for authenticating the server. The verification method that is currently used the most by CAs is the Domain Control, Validation (DCV) after the Certificate Signing Request (CSR) is submitted to a PKI/CA. [8] This mechanism includes many steps done manually, for configuration and validation of the domain name. Let's Encrypt has proposed a simplified method to use of the HTTPS protocol on the server side by the adoption of the ACME method and protocol. This method automates the server the certificate issuance and validation and therefore provides certificates at low cost. The ACME has improved the automation but not solved validation. We reviewed several research works focusing on using the Blockchain technology to solve the trust issues of PKI/CA and the problem of key management. Yakubov et. al proposed a Blockchain-based PKI framework to manage X.509 certificates [9]. Their main contributions include embedding metadata into the X.509 extension fields and providing the certificate revocation mechanism. This proposed system is hybrid as it combines the use of existing X.509 certificates with Blockchain technology. it has some major drawbacks: (1) Increased complexity added to the existing PKI infrastructure. (2) Modification of standard certificates to include Blockchain metadata in their extension fields. (3) Centralization, as certificate authorities are still required for the proper functioning of the system. (4) Assignment of smart contracts for CAs: each CA is required to have its own smart contract. Fromknecht et al. proposed Certcoin [10], a decentralized PKI based on Bitcoin's Blockchain, giving the ability for users to register a domain, update a public key pertaining to a domain, verify and look up for a public key, and key revocation. Certcoin aims at combining the Pretty Good Privacy's (PGP) Web of Trust (WoT), and the Blockchain technology. However, Certcoin lacks the validation and the authentication of the claimed domains, which makes it easier for attackers to impersonate legitimate ones.

A. Blockchain Background

The blockchain was developed for the main purpose to exchange money transactions without the need of a third party which resulted in a new digital currency called Bitcoin. Later researchers started to analyse the blockchain technology and the advantages to be a used as a distributed systems. This resulted in developing based on the blockchain concept a generic platform to run application software in a distributed way over all the blockchain nodes in a network. The application software running over these nodes are called "smart contracts" in case of Ethereum. a new term was introduced which is Distributed Applications (DApps) where the its program is called " Smart Contracts". A smart contract is defined as a program code running over Blockchain network and nodes, executing the Application software in a decentralised way without the need of centralized servers. The main benefits over the existing program execution are as follow: (1)Autonomous: the software execution is run simultaneously on all the nodes of the network.(2) Trust-less: the ledger's version is validated based on consensus algorithm among nodes.(3) Immutable data: the application's data remain permanently and unmodified in the Blockchain. (4) Transparency: smart contract's code and Data are publicly available [11].

B. Blockchain network

Blockchain can be considered as a chain of blocks that are connected to each other and duplicated in each node on the network. Each block contains mainly certain number of transactions which are stored after verification throughout the network. The verification is done via a consensus between the nodes. The information in Blockchain cannot altered or modified be added once they are approved which makes the blockchain as fraud-proof. The consensus algorithm adopted in Bitcoin and Ethereum 1.0 is the Proof-of-Work (PoW) which energy consumption as it is based on the mining principle. The PoW requires a blockchain node to compete with the others nodes in the network to solve an intensive mathematical puzzle which requires high processing power. In Ethereum 2.0 another consensus algorithm is adopted which is the Proof-of-Stake (PoS). The integrity of the stored transactions' data is preserved among all nodes. (PoS) selects a validating node based on the wealth and the age of its stake. Users inter work with the Blockchain network after creating its own wallet. It contains mainly a pair of the generated user public/private keys. The role of the private key to sign transactions (e.g., transfer digital asset from from one account user to another to another use. The public key represents the address of the wallet and verify the transactions performed by other users.

C. Ethereum

Ethereum is a Blockchain-network where their nodes are mainly a distributed computing platform that executes the same software called "smart contract". It includes the open source machine called Ethereum Virtual Machine (EVM) as its run time environment. The smart contracts run on EVM are written in high-level programming languages like Solidity

and lately other programming language like Java and Python. This High level software are compiled to EVM bytecode for execution[11]. Executing smart contracts functions is considered as a transaction in Ethereum. The cryptocurrency which Ethereum uses is called Ether. It is not only transferred between the users, but also to incentivize miners who run the Blockchain network [12]. As the EVM instruction's execution consumes the nodes processing and memory resources, a transaction fee is payed to be able to execute the contract's function. Gas consumption depends on the calculations done by the miners to execute transactions; the more complex the transaction or the operation, the more Gas needed. A Blockchain can be build in two different modes: public and private called permissioned.

D. Ethereum 2.0

The Ethereum 2.0 is gradually deployed and replacing the Ethereum 1.0 according to the time frame by the end of 2022. Ethereum 1.0's main problem

1) The speed of execution of a transaction. In Ethereum 1.0 can only process approximately 15 TPS (Transactions Per Second) meanwhile the Ethereum 2.0 should reach up to 100,000 TPS.
2) The cost of the transaction fees which is increasing continuously due to the (gas price) related the Ether value.
3) The performance issue of Ethereum network to properly handle the surge in activity.

On another hand, Ethereum 2.0 reduce high energy usage by abandoning the mining in the proof-of-stake consensus and the large disk space requirements for nodes (fixed by sharding). The main features of the Ethereum 2.0 are

1) the proof-of-stake (PoS): It is a consensus protocol which is replacing the compute-heavy (proof-of-work system based on the mining. PoS brings defense as well against 51% attacks.
2) Chains sharding: A method to divide the main blockchain into a number of "smaller" blockchain that can work in parallel, which results in a faster transaction execution and therefore make the overall system way more efficient [13],[14],[15].

III. THE PROPOSED SOLUTION

A. Blockchain-based Domain Control Validation method

In this section the proposed system is described with a detailed explanation of its components. The Ethereum blockchain network is used as a key-store that accepts and save the public keys of the servers, mapped to their identities [15],[16],[17]. The domain verification process relies on the use of the trustless Blockchain, which is a decentralized network. The system proposal consists of developing a mechanism for domain ownership verification using the Ethereum network, similar to the PKI scheme that performs on-chain verification, with minimal trust. A server is a device with an addressable identity, the ownership verification of the claimed identity is critical for authenticating the server. After the sever certificate has been created and validated by the DCV manually or by ACME, the new added feature is to validate the domain certificate using the Ethereum Blockchain network. The system architecture is depicted in Fig 1. Each device (server) having a Ethereum wallet address, registers a domain by providing a domain name, public key, the device's static public IP Address and the expiry date of the certificate. After that a verification process must occur to confirm domain authenticity. The verification is achieved by running on the Ethereum nodes a new developed smart contract and interfacing the Internet through Provable^{TM} smart contract. Provable provides "the Random Datasource" [18] and enables delivery of on-demand unpredictable data by a non trusted third-party, without risks of front-running or tampering. The aim of a service like Provable's Random Datasource is to be completely outside the trust equation, by providing cryptographically binding data authenticity proofs by widely recognized actors, leveraging the attestation technique. Based on Provable's white-paper [18] a new data source was made available by the provable API that can securely deliver entropy to the contract. An example of the Provable design was described in [18], where the external source was a Trusted Execution Environment (TEE). The Provable in this case prevents tampering the Data and protects the user from several attack vectors.

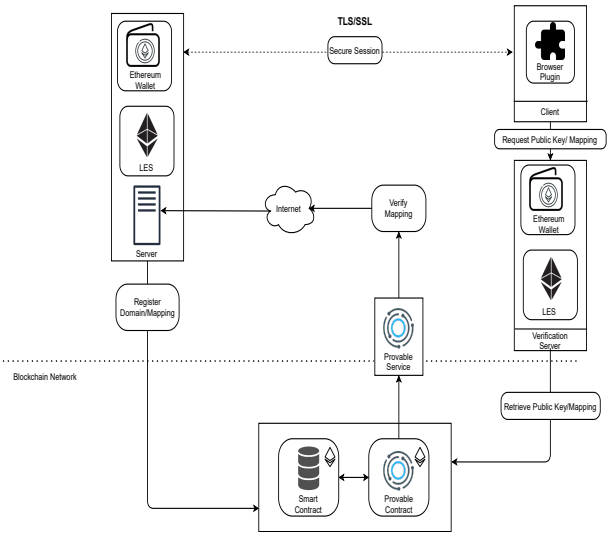

Fig. 1. B-DCV System Architecture

The authenticity proof, attached with the result, can be verified off-chain and when they are received by any "smart contract". The random data source is leveraging the Ledger proof. The principle of this method of securely feeding off-chain randomness into the blockchain is explained in the "A Scalable Architecture for On-Demand, Untrusted Delivery of Entropy" white paper [18].

A server can generally perform different type of function protocols like: HTTP, SIP, Constrained Application Protocol

(CoAP), Message Queuing Telemetry Transport (MQTT), and others. Since the server has an addressable identity, a verification process must take place before registering its domain with the smart contract.

Domain Registration "registerDomain" **Domain Data structure in the smart contract** The smart contract running on Ethereum maps the domain names to their Domain data structure. The domain struct holds multiple variables relevant to a specific domain: These structs are generated when the *"registerDomain"* function is called with the following parameters: Domain name, Public key, IP Address, date (in unix time) It should be noted that a condition must be satisfied to complete the registration; domain name provided must not be already registered. Generated structs are saved inside a map using the domain name as key. *Domain information "getDomain"* Domain information can be received after calling *"getDomain"* and passing it the domain name as parameter. The smart contract contains functions for the identities' verificationand interfacing Provable.

The software components of the system are the following: (1) The Domain owner software in the server side implementing the B-DCV function. (2) Smart contract software interfacing the Provable in the Ethereum blockchain.

We intend to make the Blockchain network apply the verification process as described below and in the sequence diagram depicted in Figure 2. First of all the domain owner generates a certificate according to the standard methods of today and install it on the webserver. The certificate can be self-signed, or optionally signed by a trusted CA. Subsequently, the webserver communicates with smart contract and sends the domain name (identity) and the generated public key. The smart contract next step is to challenge the domain owner to prove its authenticity. The smart contract requests Provable to generate a Random number (N1), the reason behind it that generating random numbers in a deterministic machine (EVM) is not possible. The domain owner, reads the N1 provided by the smart contract and generates a random number N2, and stores H(N1$\oplus$ N2) into the webserver's root directory (.well-known/pki-validation), where H is a hashing function. Other functions could be considered and adopted, but in this paper we have described the method adopted in our proof of concept. Afterwards, it sends N2 to the smart contract which request from the server by using (HTTP GET) the file hosted on the root directory. When the file is received in the smart contract, it calculates the hash of the received with the ones saved in the smart contract. It verifies the Hash, the result's proof and approves the server's record. See the sequence diagram Fig 2. *Domain configuration modification "modifyDomainConfig"* Added function called *"modifyDomainConfig"* that enables only the domain owner (same wallet address) to modify that domain configuration public key and IP address. Having the domain IP address included in the domain struct enables clients to verify the DNS server they are using has resolved the domain name to the correct IP address. This acts as an additional security layer in case of a DNS cash poisoning, where attackers replace a legitimate IP address by a malicious one, clients can be redirected to the correct address.

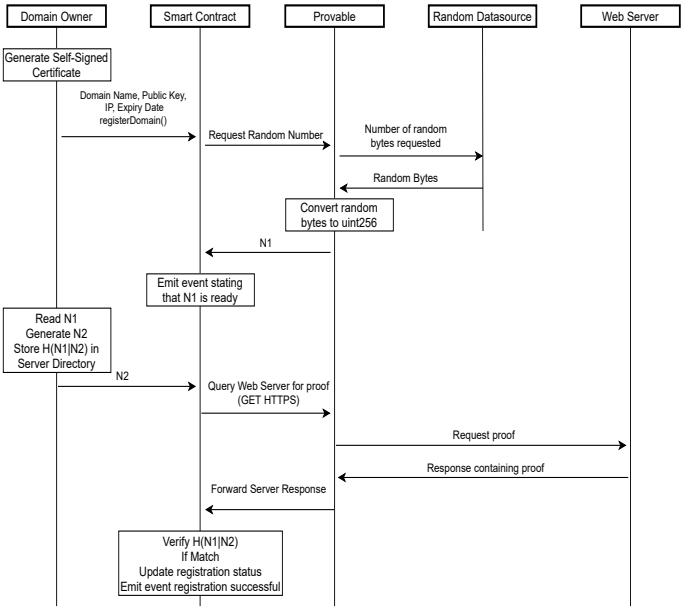

Fig. 2. Domain Control Verification

Domain certificate revocation "revokeCertificate" Calling *"revokeCertificate"*, would set the *"isValid"* flag to false letting clients know that the domain has an invalid certificate.

B. A secure Client-Server Session Establishment

This section describes an enhanced method to establish a secure session between a client and a server using the Transport Layer Security (SSL/TLS/DTLS). This method is based on the B-DCV and the intention to extend the trust and the authentication of a server side combining the standard CA method with the sever public keys stored in the Ethereum blockchain. The SSL/TLS protocol session establishment remains the same between the client and the server with a simple added optional modification at the end of the end of the session establishment. The modification is backward compatible with the existing systems (i.e., web server based on PKI/CA). The messages exchanged between the client and the server for the session setup is explained in the sequence diagram Figure 3. The client compares the public key stored in the Blockchain against the certificate's public key stored in the webs browser for example through the standard SSL/TLS protocol. The client communicates with a dedicated server called "verification server" to get the webserver public key stored in the smart contract. The role of "Verification server"

is to interface the Ethereum by retrieving the domain structure stored in the smart contract. The plugin on the client side mainly the web browser. This is developed as plug in software in major a web browser.

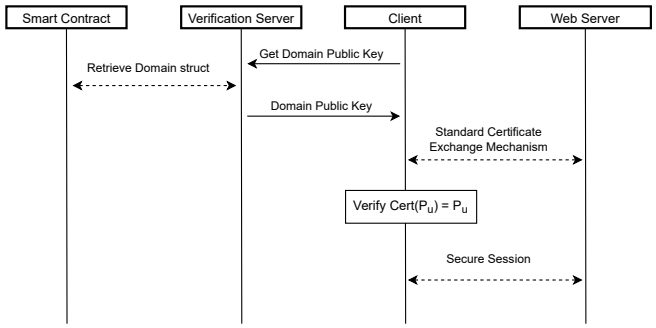

Fig. 3. Client-Side Domain Verification

It should be noted that our implemented solution is evaluated on the Rinkeby testnet where the PoA consensus is applied which is an improved version of the PoS.

IV. IMPLEMENTATION AND RESULTS

Testing was done using Remix IDE with an injected web3 environment provided by MetaMask [19]. Smart contracts are programmed using Solidity as a programming language. Remix [20] is an IDE used to write, compile and debug Solidity code. MetaMask is a software cryptocurrency wallet used to interact with the Ethereum blockchain It allows users to access their Ethereum wallet through a browser extension or mobile app, which can then be used to interact with decentralized applications. The monitoring tools used to track the transactions and queries:EtherScan [21], Provable check query status tool [22]

The deployment of the contract and the evaluation was performed on the Rinkeby testnet applying the Ethereum 2.0 based on a Proof of Authority (PoA) consensus algorithm. The test consists of simulating different transactions while recording their costs. The (PoA) is a modified form of Proof of Stake (PoS) where instead of stake with the monetary value, a validator's identity performs the role of stake. The readings obtained are assumed to be close to those we would get while using the main net on Ethereum 2.0. The provable query which is a function like other smart contract and has a base price in gas. When the Provable query gets called from another contract, it needs Ether has to execute the sending the callback transaction back to the calling smart contract. The provable query automatically recovers the fee at execution time. The fee consists of two parts [23] :

- The amount of Wei for the data source and the authenticity proof requested, converted to the recent exchange rate to US dollars price.
- The amount of Wei needed by Provable for sending the callback transaction

TABLE I. PROVABLE PROOF CALL FEES IN USD

		Proof Type			
Datasource	Base price	None	TLS Notary	Android	Ledger
URL	0.01$	+0.0$	+0.04$	+0.04$	N/A
WolframAlpha	0.03$	+0.0$	N/A	N/A	N/A
IPFS	0.01$	+0.0$	N/A	N/A	N/A
random	0.05$	+0.0$	N/A	N/A	+0.0$
computation	0.50$	+0.0$	+0.04$	+0.04$	N/A

TABLE II. UTILIZATION FEES

Functions	Gas	Provable Ether Used	Pending Transaction Time (seconds)
Contract Deployment	5384627	N/A	8.38
Register Domain	336063	N/A	8.64
Generate Random Number (Ledger Proof)	147095	0.0041765	4.44
Modify Domain Configuration	44033	N/A	14.21
Send HTTPS Request + Response	202698	0.0040353	13.71

Table II provides the utilization fees per function for the domain verification transactions B-DCV. Including the estimation of the Provable ether used: Generate the random number and sending the HTTPS request and response (the callback cost). Table III provides the total Domain Control Verification cost.

TABLE III. TOTAL CURRENT RATE FEES

	Gas(Gwei)	Provable cost(Eth)	Total
Domain Control Verification cost	729889 = 1.726$	0.00821=19.4483$	21.1743$

As of writing this paper around October 2021, 1 Eth is equal to 2,368 US dollar.

V. COMPARISON WITH THE EXISTING CERTIFICATES ASSIGNMENT AND VALIDATION

A. certificates assignment and validation based on PKI/CA

The tables below provide an insight on the different types of certificates and certificate providers pricing as of October 2021.

TABLE IV. CERTIFICATE TYPES

Certificate type	Issuance delay	Condition	Target
Domain Validated	Within minutes	Verify ownership of domain to vendor	Websites without sensitive data
Organization Validated	1-3 days	Verify domain and organization with CA	Websites without customer data
Extended Validated	3-4 days	Verify domain, provide personal and organizational information to CA	Websites with customer data

B. The (B-DCV) fees and features compared to the PKI/CA

- Onetime fee of USD 21.2 per domain (No need to renew) as it is today without taking into consideration the Layer2 Ethereum
- Instant issuance and Use self-signed certificates
- Having the public IP address of the domain available on demand in the smart contract, makes attacks that redirects users to fake websites, like SSL stripping and man in the middle attacks, easily noticeable and mitigated from the client's side.
- Certificate Expiry date available on demand and can be checked by the users thus increasing security and preventing man-in-the-middle attacks

TABLE V. CERTIFICATE PROVIDERS

Certificate Provider	Certificate Price (USD)
Digicert [25]	238/yr
SSL.com [26]	49/yr
Namecheap [27]	19.55/yr
Renews	28.88/yr
Comodo ssl store [28]	28/yr
Global sign [29]	249/yr
Thawte [30]	238/yr

C. Layer 2 Ethereum

As the number of transactions on Ethereum has increased, the blockchain has reached its capacity limit which has caused an increase of the network usage fees. Due to that there was a need for scaling solutions. The main objective of scalability is to increase transaction speed and throughput and at the same time reducing the transaction fee, while maintaining decentralization and security. The scaling solutions are implemented separately from layer 1 main network and they require no changes to the existing Ethereum protocol. The "layer 2" solutions, inherit their security directly from layer 1 Ethereum consensus, such as rollups or state channels. Other solutions communicate with the main network, but derive their security differently to obtain a variety of goals [24]. The fees of the B-DCV will drop dramatically compared to the existing cost when the Ethereum layer 2 is adopted in the proposed solution.

VI. CONCLUSION AND FUTURE WORK

This paper proposes a new method for the sever certificate validation and verification based on the Ethereum Blockchain. The focus in this work was as well on the implementation and the development of the smart contract, the server verification, and the plugin software in the web browser. The solution was tested on Rinkeby testnet using a modified version of the Ethereum 2.0 which gave promising results related to the cost and speed. We aim as a future work to present this concept in more respected forums supported by more testing results and form a group to push for the standardization of this protocol. We believe that this method will ultimately resolve the security issues generally in the internet and mainly for the client-server communications.

REFERENCES

[1] R. Barnes, J. Hoffman-Andrews, D. McCarney, and J. Kasten, "AutomaticCertificate Management Environment (ACME)," RFC 8555, RFC Editor, March 2019.

[2] Implementation state of HSTS and HPKP in both browsers and serverss S de los Santos, C Torrano, Y Rubio, Conference on Cryptology 2016 - Springer.

[3] D. Fisher, "Final report on diginotar hack shows total compromise ofca servers," Retrieved September, vol. 8, p. 2013, 2012.

[4] N. Leavitt, "Internet security under attack: The undermining of digital-certificates," Computer, vol. 44, no. 12, pp. 17–20, 2011.

[5] S. Gangan, "A review of man-in-the-middle attacks," arXiv preprint arXiv:1504.02115, 2015.

[6] Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL Christopher Soghoian and Sid Stamm.

[7] E. F. Kfoury, D. Khoury, A. Alsabeh, J. Gomez, J. richigno, and E. Bou-Harb, "A Blockchain-based Method for Decentralizing the ACME Protocol to Enhance Trust in PKI," 2020 43rd Int. Conf. Telecommun. Signal Process. TSP 2020, no. May, pp. 461–465, 2020, doi: 10.1109/TSP49548.2020.9163555

[8] "Nexcess"https://help.nexcess.net/ssl/what-is-a-cs

[9] A. Yakubov, W. Shbair, A. Wallbom, D. Sanda, et al., "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, Tapei, Tawain 23-27 April 2018, 2018.

[10] C. Fromknecht, D. Velicanu, and S. Yakoubov, "Certcoin: A namecoin based decentralized authentication system," Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep, vol. 6, 2014.

[11] N. Szabo, "The idea of smart contracts," Nick Szabos Papers and Concise Tutorials, vol. 6, 1997.

[12] Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum project yellow paper, vol. 151, pp. 1–32, 2014.

[13] "Ethereum 2.0"https://ethereum.org/en/eth2

[14] "Combining GHOST and Casper"https://arxiv.org/pdf/2003.03052.pdf

[15] E. Kfoury and D. Khoury, "Distributed Public Key Infrastructure and PSK Exchange Based on Blockchain Technology," Proc. - IEEE 2018 Int. Congr. Cybermatics 2018 IEEE Conf. Internet Things, Green Comput. Commun. Cyber, Phys. Soc. Comput. Smart Data, Blockchain, Comput. Inf. Technol. iThings/Gree, pp. 1116–1120, 2018, doi:

[16] E. F. Kfoury and D. J. Khoury, "Secure end-to-end VoIP system based on ethereum blockchain," J. Commun., vol. 13, no. 8, pp. 450–455, 2018, doi: 10.12720/jcm.13.8.450-455.

[17] E. F. Kfoury, J. Gomez, J. Crichigno, E. Bou-Harb, and D. Khoury, "Decentralized Distribution of PCP Mappings over Blockchain for End-to-End Secure Direct Communications," IEEE Access, vol. 7, no. August, pp. 110159–110173, 2019, doi: 10.1109/ACCESS.2019.2934049

[18] "A Scalable Architecture for On-Demand, Untrusted Delivery of Entropy" https://provable.xyz/papers/random_datasource-rev1.pdf

[19] "MetaMask Wallet"https://metamask.io

[20] "Remix IDE"https://remix.ethereum.org/

[21] "EtherScan"https://rinkeby.etherscan.io

[22] "Provable query status check tool"https://app.provable.xyz/home/check_query

[23] "Provable call fees in usd"https://docs.provable.xyz/#pricing-advanced-datasources-call-fee

[24] https://ethereum.org/en/developers/docs/scaling

[25] https://order.digicert.com/step1/ssl_basic?validity=3

[26] https://www.ssl.com/certificates/basicssl/buy/

[27] https://www.namecheap.com/security/ssl-certificates/

[28] https://comodosslstore.com/essentialssl.aspx

[29] https://shop.globalsign.com/en/ssl-tls-certificates

[30] https://www.thawte.com/ssl/