Identifying digital certificate revocation using Bitcoin

Cristòfol Daudén-Esmel, Alexandre Viejo
and Jordi Castellà-Roca
Universitat Rovira i Virgili
Dept. d'Enginyeria Informàtica i Matemàtiques, CYBERCAT
Av. Països Catalans 26, E-43007 Tarragona, Spain
{critofol.dauden, alexandre.viejo, jordi.castella}@urv.cat

Jordi Herrera-Joancomartí
Universitat Autònoma de Barcelona
Dept. d'Enginyeria de la Informació i les Comunicacions
08193 Bellaterra, Catalonia, Spain
jordi.herrera@uab.cat

***Abstract*—The Internet has transformed global communication, daily routines, and business practices by facilitating vast data exchanges, much of which is sensitive and requires protection. Public Key Infrastructure (PKI) is the most widely adopted solution for addressing these security needs. PKI provides a framework for managing public-key cryptography through the issuance, management, distribution, and revocation of certificates that bind public keys with entity identities. A critical aspect of PKI is certificate revocation, which invalidates compromised or fraudulent certificates to prevent security breaches. Various methods, including Certificate Revocation Lists (CRL), Online Certificate Status Protocol (OCSP), and Certificate Revocation Trees (CRT), have been developed, each facing challenges like scalability, cost, and privacy concerns. In this work we propose an enhancement to the traditional PKI infrastructure by introducing a lightweight certificate revocation and validation mechanism that improves availability and privacy. The proposal features two revocation approaches: unilateral and bilateral. Unilateral revocation allows either the user or the Certificate Authority (CA) to revoke the certificate, predetermined during certificate generation. Bilateral revocation permits both the user and the CA to revoke the certificate, with further distinctions on whether one party can act alone or both must cooperate.**

***Index Terms*—PKI, certificates, X.509 standard, UTXO**

I. Introduction

The Internet has revolutionized the way people live, communicate, and do business worldwide. Today, it's an indispensable part of our daily routines, facilitating vast amounts of data exchange. Much of this data is sensitive and requires safeguarding to protect users' privacy. Moreover, many communications and services necessitate authentication for access. Addressing these security needs, various mechanisms have been proposed, with the Public Key Infrastructure (PKI) emerging as the most prevalent solution.

PKI encompasses a framework of authorities, policies, and procedures for managing public-key cryptography. It binds public keys with entity identities through registration and certificate issuance processes. PKI manages, distributes, and revokes certificates crucially, ensuring secure transactions and communications.

Certificate revocation is essential in PKI security, allowing the invalidation of compromised or fraudulent certificates before their expiry. Failure to revoke such certificates can have severe consequences, as seen in incidents like the Heartbleed SSL/TLS vulnerability, where thousands of certificates were revoked to mitigate risks posed by attackers exploiting vulnerable servers [1] [2]. Without proper revocation mechanisms, attackers can impersonate legitimate sites, leading to data breaches and malware infections.

In order to address this issue, various revocation methods have been proposed, including Certificate Revocation Lists (CRL), Online Certificate Status Protocol (OCSP), and Certificate Revocation Trees (CRT). However, each method presents its own set of challenges, such as scalability issues, financial costs, and privacy concerns [3]. Finding a balance between effectiveness and efficiency remains an ongoing challenge in PKI security.

A. State of the Art

Considering the drawbacks of revocation methods currently used with the X.509 standard [3] and with the arise of blockchain technology, we find numerous proposals in the literature to design and create effective and secure solutions. These efforts often revolve around the design aspects concerning the decentralization, integrity, and non-repudiance of blockchain's technology. Based on these criteria, BC based certificate revocation methods can be categorized into two main categories: i) new certificate standard proposals, and ii) X.509 standard build-in-top proposals.

The first category, *new certificate standard proposals*, primarily focuses on the usage of BC technology to store the new designed certificates and perform all revocation process in the BC itself. Some relevant proposals that fall within this category are [3]–[9]. However, all proposals in this category can be difficult to accept due to the well established and usage of the current X.509 standard.

In the solutions that fall into the second category, X.509 standard build-in-top proposals, extension field of the X.509 standard along with BC technology is used for certificate revocation. Noteworthy schemes within this category are [10]–[14]. Given their direct alignment with the requirements outlined in this paper, these are next discussed in further detail.

In [10], authors describe SCRaaPS (Secure Certificate Revocation as a Peer Service), a method to support X509 certificate revocation using the Scrybe provenance system. Scrybe along with the use of a Cuckoo filter, replaces OCSP servers by storing the serial number of the revoked certificates along with the entity's public key and the CA's public key. However, there is not an implementation of the system neither they explain who maintains the Scrybe blockchain architecture.

Additionally, this proposal does not allow users self revocation of the certificates.

Authors in [11] propose a decentralized digital certificate revocation system based on a consortium BC. On this scheme multiple CAs collaborate in the management of a distributed CRL and introduce s secret sharing scheme to improve the fault tolerance and ensure the reliability of the maintenance process. This proposal is not compatible with current X509 standard implying an overhead for the CAs in the revocation process and in the maintenance of the full ecosystem.

Yan et. al propose in [12] a PKI system based on the BC technology. On this architecture, a user generates a self-signed certificate and submits it into the BC through a submission node. This node is the responsible of verify and endorse the certificate. When a user needs to revoke a certificate, she changes the status of that certificate in the ledger to "revoked". This proposal is not compatible with current X509 standard and they do not evaluate the economic costs associated to the storage of the certificates on the BC.

The authors in [13] and [14] propose a certificate revocation and status revocation scheme that leverages on the blockchain technology. In their solution, they use the extension field of the X509 certificate's structure to introduce a field that describes which distribution point the certificate will belong to if revoked. These points are presented by a Bloom filter stored in the BC with the revoked certificates on it. The main issue with this proposal is the extra overhead and costs on the CAs. Each time a certificate is revoked, the CA has to re-compute the corresponding Bloom filter and perform a new transaction to store it on the BC. Additionally, servers must continuously check new transactions performed by its CA, thus checking the transactions related to its distribution point.

B. Contribution

This work tackles the challenges previously outlined by presenting a comprehensive solution: an improvement of the traditional PKI infrastructure by introducing a lightweight certificate validation mechanism.

In essence, our approach consists of the usage of an unspent UTXO which identifier is included in the *utxoID* custom extension filed of the X.509 standard. In order to revoke the certificate, the UTXO output is spent by creating and broadcasting a transaction using a standard Bitcoin wallet. The proposal includes two revocation approaches: unilateral and bilateral. In the unilateral revocation approach, either the user or the Certificate Authority (CA) can revoke the certificate. On the other hand, the bilateral revocation approach allows both the user and the CA to revoke the certificate, with variations on whether one party can act independently or if both must cooperate.

The rest of this paper is organized as follows: Section 2 offers a brief overview of the technologies and concepts used in our contribution. Section 3 formalizes the protocols that comprise our solution. Section 4 assesses the compliance with flexibility, scalability, privacy and cost requirements. Finally, Section 5 presents the concluding remarks.

II. Background

A. PKI background

PKI is a framework that facilitates secure communication over the internet by employing asymmetric encryption and digital certificates. These certificates, issued by trusted entities known as Certification Authorities (CAs), verify the identities of websites, servers, and individuals, thereby establishing secure connections and safeguarding sensitive information from unauthorized access or tampering. However, as with any technology, vulnerabilities can arise, and certificates may be compromised or become invalid over time. This is where the crucial concept of Certificate revocation comes into play. Certificate revocation is the process by which certificates that are no longer trustworthy or valid are flagged and rendered unusable, safeguarding users from potential security breaches and ensuring the integrity of online interactions. Current certificate revocation methods are Certificate Revocation Lists (CRL), Certificate Revocation Trees (CRT) and Online Certificate Status Protocol (OCSP).

The Certificate Revocation List (CRL) [15] is a record maintained by a Certification Authority (CA) containing details of revoked certificates. It is dated and signed by the CA and is periodically published. To verify the validity of a certificate, a requester must send a request to the publication server hosting the corresponding CRL, specifying the CA's identifier. The requester receives the latest CRL from the CA, checks its signature and validity, and then searches for the certificate in the CRL. The CRL offers simplicity, comprehensive information, and low risk. However, its major drawback is its size, which requires high bandwidth for updates and verification, limiting its scalability.

The Certificate Revocation Tree (CRT) [16] typically takes the form of a Merkle hash tree within a PKI domain. It encapsulates all certificate revocation data, with each leaf containing statements about certificate serial numbers. The leaves are logically ordered by serial number, with adjacent certificates indicating revoked status. This structure efficiently determines whether a certificate is revoked or not, with the path from the root to the leaf providing proof of a certificate's status. The CRT offers the advantage of not requiring the entire Certificate Revocation List (CRL) for verification. However, its weakness lies in update management, as any change in the set of revoked certificates necessitates re-computation of the entire CRT, resulting in a continuous workload.

The Online Certificate Status Protocol (OCSP) [17] is an online revocation system using a request/response mechanism, with revocation information hosted on an OCSP responder server, directly from the CA. OCSP exclusively checks revocation status and relies on third-party trust, as responses are not signed by the CA, requiring trust in the responder server. End entities send OCSP requests to verify certificate status, receiving signed OCSP responses in return. This approach improves timeliness and revocation information updates but faces drawbacks such as a centralization poses a single point of failure, lacks serial number and CA validation, costly in time and latency, ineffective for offline systems, potential lag in updated revocation information, and privacy risks due to responder tracking user activities.

B. Blockchain background

A blockchain is a decentralized and distributed digital ledger that records transactions across a network of computers. Each transaction is verified and stored in a block, which is linked to previous blocks, forming a chronological chain of blocks given its "blockchain" name.

The blockchain landscape comprises two primary models: the account-based model and the UTXO-based model [18], each offering distinct approaches to managing and recording transactions. In the account-based model (like Ethereum [19]), the state of the blockchain is represented by the account balances of participants, with transactions involving the transfer of value between these accounts. Conversely, the UTXO-based model (like Bitcoin [20]) revolves around unspent transaction outputs, where transactions consume specific inputs and create new outputs. While both models share the common goal of facilitating decentralized transactions, they differ in their data structures, transaction handling, and scalability approaches, catering to diverse blockchain use cases and preferences.

The UTXO-based blockchain model provides different benefits regarding efficiency and privacy. UTXO allows for more parallel processing compared to account-based models. In UTXO, each transaction consumes specific outputs (UTXOs) and creates new ones, which means transactions can be processed independently without needing to worry about conflicting updates to shared account balances. This parallelism can lead to higher throughput and scalability. Furthermore, UTXO-based systems can often simplify verification processes. Each transaction simply needs to prove that it is consuming valid outputs (UTXOs) and creating valid ones. This simplicity can make verification faster and less resource-intensive, especially in large-scale networks. Another benefit of the UTXO-model is that it reduces the state sizes of the blockchain compared to account-based models. In UTXO, the state is represented by the set of unspent transaction outputs, which tends to be more compact compared to storing account balances for every user. Smaller state sizes can lead to faster synchronization, lower storage requirements, and improved overall performance. Finally, UTXO model inherently provides better privacy features since each UTXO is independent from other UTXO and, although they may be linked using clustering techniques, there exists privacy-enhancing techniques to avoid such linkability thread.

Of course, depending on the application, account based blockchains may be more suitable since they allow to create more complex and diverse smart contract functionalities. However, regarding the goal of our proposal, the simplicity of a UTXO based blockchain suits perfectly for our purpose, so we will base our proposal in such a blockchain model.

III. Proposal

As we already mention, our proposal is based on a baseline PKI infrastructure from which we want to improve the certificate validation process, providing a lightweight mechanism that provides better availability and privacy properties. To that end, we assume the existence of the following parties in our architecture:[1]

- Certification authority (CA):
- Validation authority (VA):
- End users (U):

Our scheme can be divided in three different protocols:

- Certificate generation
- Certificate validation
- certificate revocation

Our proposal presents two different approaches: unilateral revocation or bilateral revocation. By unilateral revocation, we mean that only one of the participants, the user or the CA, can revoke the certificate and the one that has this ability has to be determined beforehand in the certificate generation protocol. In the bilateral revocation, both the user and/or the CA can revoke a certificate. In the latter approach, we can distinguish whether one of the two users can revoke the certificate or if both of them are needed and have to cooperate in the revocation process.

It is worth mention that the standard PKI revocation mechanism falls inside one of the cases of the unilateral revocation approach, since only the CA can revoke a certificate.

In the next subsections, we are going to detail each of the two above-mentioned protocols for each of the two approaches.

A. Unilateral revocation approach

1) Only the user can revoke: In this approach, the definition of the three protocols of our scheme is as follows:

The **certificate generation** procedure has the following steps:

1) User U generates a private/public key pair.
2) U selects a UTXO in his wallet, defined by a transaction identifier, tx_{id}, and an output id, $vout$. Notice that, as we will see later, the exact value in bitcoins of such output is not relevant for the correctness of the proposed protocol.
3) U creates a certificate signing request CSR. In such CSR, in a custom extension filed *utxoID*[2], U includes the identifier of the UTXO generated in the previous step, that is tx_{id} and $vout$.
4) U sends the CSR to the certification authority.
5) CA validates that the information included in the CSR is correct, and particularly, verifies that the identification of the UTXO provided has not been spent and that the UTXO provided can be spent by the U, ensuring that U may revoke the certificate.[3]
6) CA generates the digital certificate using the standard procedure.
7) CA sends the digital certificate to U.

The **certificate validation** procedure has the following steps:

1) Any entity may present the certificate in front the VA.

[1]These are the minimal entities that interoperate in our proposal. However, other standard PKI entities, like a Registration Authority, can be added in the scheme without any modification.

[2]Inserting Custom OIDs into OpenSSL: https://knowledge.digicert.com/quovadis/ssl-certificates/csr-generation/inserting-custom-oids-into-openssl

[3]The validations about the output can be omitted in case the CA does not care about the possibility to generate an already revoked certificate or does not care about who is able to revoke the certificate.

2) VA extracts the transaction ID, tx_{id} and the output id, $vout$, from the certificate and queries the blockchain whether such output is still spendable or not.
3) If the output is spendable, the certification is valid. In case the output is already spent, that means the certification has been revoked and is no longer valid.

The **certificate revocation** procedure has the following steps, performed by the user who has the ability to revoke the certificate:

1) U extracts the transaction ID, tx_{id} and the output id $vout$ from the certificate.
2) U spends the output by creating and broadcasting a transaction with such output using a standard Bitcoin wallet.

2) Only the CA can revoke: This approach is completely equivalent to the previous one. The minimum difference is that in the **certificate generation**, the CA is the one who includes the transaction identifier, tx_{id}, and an output id, $vout$, in the certificate before generation, including an UTXO that she owns. Furthermore, in the **certificate revocation** process, is the CA the one that has to spend the output in order to revoke the certificate.

B. bilateral revocation approach

1) Either the user or the CA can revoke indistinctly: The **certificate generation** procedure has the same steps that those of Section III-A1 except for the step 2 in which the UTXO needed has to belong to both, the user and the CA. To that end, the features of the scripting bitcoin language are used to obtain an UTXO locked with a (1-2) threshold scheme, where one key belongs to U and the other key belongs to the CA, and only one of the two keys is needed to spend the UTXO. Notice that the process to obtain such UTXO can be initialized either by the user or by the CA and the final value of the transaction ID, tx_{id}, and the output id, $vout$, can be included in the CSR by the user or in the final certificate by the CA.

The **certificate validation** procedure is exactly the same as the one described in Section III-A1.

Finally, the **certificate revocation** procedure is the same as the one described in Section III-A1 but it can be performed indistinctly by the user or by the CA, since both of them are able to spend the UTXO identified in the certificate.

2) Both the user and the CA have to collaborate to revoke: This approach is analogous to the approach of the previous section with some minor details.

In the **certificate generation** procedure, the UTXO needed has to belong to both, the user and the CA. To that end, the features of the scripting bitcoin language are used to obtain an UTXO locked with a two-multisignature scheme, that is a scheme with two keys, one belonging to U and the other to the CA, in which both keys are needed to spend the UTXO. As in the previous key, the process to obtain such UTXO can be initialized either by the user of by the CA and the final value of the transaction ID, tx_{id}, and the output id, $vout$, can be included in the CSR by the user or in the final certificate by the CA.

Finally, the **certificate revocation** procedure differs from others proposals since the user and the CA have to coordinate to spend the UTXO.

IV. Proposal properties

The lifecycle of a digital certificate goes from the generation to the expiration or revocation and, in between, its usage includes another process: the certificate validation. Notice that, while the generation, expiration or revocation of a certificate takes places a only once in its life cycle, the validation process is performed every time the certificate is used during his lifetime. So verification is the most intensive process of the CA duties.

In our proposal, we outsource the certificate validation register from the CA to the Bitcoin blockchain state, in which we have encoded the certificate validation in terms of UTXO. In doing so, we have moved from a centralized database model to a distributed one, enhancing the properties of the certificate validation process, as we describe in the next subsections.

1) Efficiency: We focus the efficiency analysis of our proposal regarding the certificate validation process since, as we mention above, such process is the one that takes place more times and consume more resources during all the certificate lifecicle, comparing to the generation or revocation.

In our proposal, to check the validity of a certificate we have to verify if the UTXO included in the certificate, specified with the transaction ID, tx_{id}, and the output id, $vout$, has been spent or not. Such verification can be performed using different mechanism.

The first one is to store the Bitcoin UTXO set locally and query such set. In fact, such an approach is straight forward if the validator has installed a full Bitcoin node. In fact, a full Bitcoin node stores the UTXO set indexed using the transaction ID, tx_{id}, and the output id, $vout$, so with a single RPC command[4] to the local node, the validation can be performed.

In case the validator does not have the UTXO set stored locally, the validator can use any Bitcoin blockchain explorer that provides information of the blockchain. In fact, such explorers has an API interface that allows the automatization of the verification process. For instance, the following API call:

https://blockstream.info/api/tx/e30e0043e2284...a7ad46096a21dd/outspend/0

returns the state of the output with number $vout = 0$ and tx_{id} =e30e0043e2284...a7ad46096a21dd.

A. Flexibility

In contrast with standard PKI revocation mechanisms, the proposed scheme allows both the CA but also the user to revoke the certificate. Such ability allows the user to revoke a digital certificate as soon as he detects that the private key may be compromised. Notice that, even in the case that the revocation takes place due to the private key lost, the revocation can be also performed by the user because the key used to revoke the certificate are related with their digital wallet and the user does not need to use the certification private key, that probably she has lost.

[4]In case the Bitcoin Core distribution is used, the following RPC call can be used `bitcoin-cli gettxout txid vout`.

B. Scalability/Resilience

The OSCP service faces scalability issues due to its resource-intensive nature, centralized control, configuration complexity, latency, and network congestion as the number of devices requesting certificate validation increases. Also, its resilience is challenged by single points of failure, difficulty in ensuring fault tolerance, significant recovery times, security vulnerabilities, and dependence on network infrastructure. Additionally, the Certificate Revocation List (CRL) used in conjunction with OSCP can be problematic. CRLs can grow large, making them cumbersome to manage and distribute, leading to delays in the revocation process and potentially allowing compromised certificates to remain active longer than necessary.

To mitigate these challenges, our solution relies on the blockchain technology, which eliminates single points of failure and improves load distribution. Blockchain's tamper-proof ledger and resistance to cyberattacks enhance security, while real-time updates and streamlined certificate management address the inefficiencies of traditional CRLs. Built-in redundancy and quick recovery mechanisms ensure fault tolerance, and transparent, auditable operations build trust.

C. Privacy

OSCP queries expose specific certificate checks, potentially revealing browsing history and enabling tracking and profiling by OCSP responders, which users must trust to handle data responsibly. CRLs require bulk data downloads, revealing security activities and necessitating frequent updates that can track behavior over time. Both methods lack anonymity, allowing servers to collect identifying metadata and correlate it with other online activities.

Our proposed solution does not have these privacy issues, as certificate validation is performed by anonymously checking if a bitcoin UTXO has been spent or not.

D. Cost analysis of the proposal

As we already mention, we use a UTXO of a Bitcoin transaction as a flag to indicate whether a certificate is valid or it has been revoked. So, in this section we want to assess the economic cost of our proposal, regarding that, in some cases, Bitcoin transactions have to be created, and such process has an associated economic cost.

The cost of creating a Bitcoin transaction can be measured with the fees needed to include such transaction in a block. Such fee is driven by the size of the data structure defining the transaction since the scarce resource in the Bitcoin network is the space in a block, since only a block is allowed every 10 minutes (in mean) and the size of such block is bounded. Transaction size is not a straight forward value, since it depends on the exact characteristics of every transaction, so we will define the type of transactions that we need, and we use standard measurements to obtain its size.[5]

Notice that our proposal interacts with the Bitcoin protocol in all three phases, namely, the certificate generation, validation and revocation.

[5]On-line tools like https://bitcoinops.org/en/tools/calc-size/ can be used to estimate a Bitcoin transaction size.

In the **certificate generation** procedure, we need an UTXO that could be an existing one or a newly created, depending on the conditions that the UTXO has in the locking script. In the *unilateral revocation approach*, the locking conditions of the UTXO shall be controlled by either the User or the CA, so any existing UTXO that the user or the CA has in his wallet can be used as a flag for the certificate. So, in that scenario, it is not needed to create a new UTXO, so no transaction is executed and there are no cost associated to that end. However, in the bilateral approach, the UTXO has to be controlled by both, the user and the CA, so a new UTXO is needed. Every new Bitcoin transaction creates new UTXOs, so a new transaction has to be created, which has a cost in fees. To assess which is that cost, we need to measure the size of such transaction, since fees in the Bitcoin network are measured based on the size of the transaction.

In order to minimize the cost needed to pay for using our protocol, we will craft the smallest possible transaction. Such transaction is a Bitcoin transaction with one input, and two outputs. Furthermore, to reduce even more the size needed in a block and so to reduce the fees, we will use the most compact address type, so the input will spend a P2WPKH output and the transaction will create a P2WPKH output and another P2WSH output that includes the multiple signature option. With such parameters, our transaction has a total size of 155 bytes, so this is the block space needed to allocate such a transaction.

In the **certificate revocation protocol**, we always need to spend the output to flag the certificate as a revoked. To do it, either the user or the certification authority needs to generate a new bitcoin transaction that spends the output indicated in the certificate and creates a new output. Again, we measure the smallest possible transaction, that will be a transaction with a single input spending the output indicated in the certificated and a single output where the funds can be transferred. The size of such transaction will depend on the output type indicated in the certificate.

In the unilateral revocation scenario, we can assume that the output we want to spend is a P2WPKH, so the transaction size will be 110 bytes. Finally, the transaction needed for the revocation in a bilateral scenario will include a single input spending a P2WSH of a 1-2 multisignature script and a single P2WPKH output that produces a transaction size of 339 bytes.

Table I summarizes block allocation costs for the transactions needed in each of the procedures of our system, depending which of the scenario (unilateral / bilateral) is chosen.

Block allocation costs	unilateral revocation	bilateral revocation
certificate generation	-	153 bytes
certificate validation	-	-
certificate revocation	110 bytes	339 bytes

TABLE I
BLOCK ALLOCATION COSTS

Once we have clearly identified such block allocation costs, we can measure the exact economic costs in terms of any fiat currency, such dollars. To do so, we need to apply the fee rate for the allocation costs, but such value is not fixed but dynamically calculated based on the saturation of the network.

To provide an estimation cost of our proposal, we take the mean fee rate[6] for the last year we have a mean fee of 45 satoshis per byte. Assuming that 1 BTC is equivalent to 10^8 satoshis and the actual rate for the bitcoin price is 67.366$ Table II provides a cost estimation for our solution.

Dollar costs	unilateral revocation	bilateral revocation
certificate generation	-	4,43 $
certificate validation	-	-
certificate revocation	3,19 $	9,82 $

TABLE II
DOLLAR COSTS

Using the system in unilateral mode without needing to revoke doesn't cost any money. Additionally, of the three processes involved (generation, validation, and revocation), validation is the most frequently performed and is free. Furthermore, the cost of 3.19 to have a digital certificate that you know doesn't track you via OCSP is a cost that users might be willing to pay for their privacy, especially considering that this amount covers the entire duration of the certificate.

V. CONCLUSIONS AND FURTHER RESEARCH

This paper introduces an enhancement of the traditional PKI infrastructure through the introduction of a lightweight certificate revocation and validation mechanism addresses key challenges in availability and privacy. By proposing two distinct revocation approaches —unilateral and bilateral— this work offers flexible solutions to improve the robustness of PKI systems.

Unilateral revocation, where either the user or the Certificate Authority (CA) has the authority to revoke a certificate predetermined during the certificate generation process, provides a streamlined and efficient method of certificate management. This approach ensures that revocation can occur swiftly, reducing the window of vulnerability when a certificate is compromised.

On the other hand, bilateral revocation allows both the user and the CA to revoke the certificate and can require cooperation between both parties, adding an additional layer of security. This method is particularly advantageous in scenarios where mutual agreement is crucial to prevent unauthorized revocations, thereby enhancing trust and security in the PKI system.

ACKNOWLEDGEMENTS

This work was partially supported by the Spanish Ministry under Grant PID2021-125962OB-C33 SECURING/NET, by the Catalan Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR) under Grant SGR2021-00643 and by the Plan de Recuperación, Transformación y Resiliencia funded with Next Generation EU funds through the project DANGER INICBE-C062/23. Additionally this research is supported by the project "HERMES" funded by INCIBE and the European Union NextGenerationEU/PRTR, by project PID2021-125962OB-C32 "SECURING/DATA" funded by MCIN/AEI/10.13039/501100011033/ FEDER, UE, and by the grant 2021SGR 00115 from the Government of Catalonia. First author is also supported by the Spanish Government under an FPU grant (ref. FPU20/03254).

[6]Data has been taken form https://privacypros.io/tools/bitcoin-fee-estimator/

REFERENCES

[1] P. Mutton, "Certificate revocation: Why browsers remain affected by heartbleed," in *Technical Report*. Netcraft, 2014.

[2] Z. Durumeric, F. Li, J. Kasten, J. Amann, J. Beekman, M. Payer, N. Weaver, D. Adrian, V. Paxson, M. Bailey *et al.*, "The matter of heartbleed," in *Proceedings of the 2014 conference on internet measurement conference*, 2014, pp. 475–488.

[3] A. Yakubov, W. Shbair, A. Wallbom, D. Sanda, and R. State, "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.

[4] M. Y. Kubilay, M. S. Kiraz, and H. A. Mantar, "Certledger: A new pki model with certificate transparency based on blockchain," *Computers Security*, vol. 85, pp. 333–352, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404818313014

[5] A. Garba, Q. Hu, Z. Chen, and M. R. Asghar, "Bb-pki: Blockchain-based public key infrastructure certificate management," in *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2020, pp. 824–829.

[6] J. Zhao, Z. Lin, X. Huang, Y. Zhang, and S. Xiang, "Trustca: Achieving certificate transparency through smart contract in blockchain platforms," in *2020 International Conference on High Performance Big Data and Intelligent Systems (HPBDIS)*, 2020, pp. 1–6.

[7] A. Gayathiri, J. Jayachitra, and S. Matilda, "Certificate validation using blockchain," in *2020 7th International Conference on Smart Structures and Systems (ICSSS)*, 2020, pp. 1–4.

[8] B. Liu, L. Xiao, J. Long, M. Tang, and O. Hosam, "Secure digital certificate-based data access control scheme in blockchain," *IEEE Access*, vol. 8, pp. 91 751–91 760, 2020.

[9] M. Toorani and C. Gehrmann, "A decentralized dynamic pki based on blockchain," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, ser. SAC '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 1646–1655. [Online]. Available: https://doi.org/10.1145/3412841.3442038

[10] S. Medury, A. Skjellum, R. R. Brooks, and L. Yu, "Scraaps: X.509 certificate revocation using the blockchain-based scrybe secure provenance system," in *2018 13th International Conference on Malicious and Unwanted Software (MALWARE)*, 2018, pp. 145–152.

[11] A. Zhang and X. Ma, "Decentralized digital certificate revocation system based on blockchain," in *Journal of Physics: Conference Series*, vol. 1069, no. 1. IOP Publishing, 2018, p. 012125.

[12] J. Yan, X. Hang, B. Yang, L. Su, and S. He, "Blockchain based pki and certificates management in mobile networks," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 1764–1770.

[13] Y. C. Elloh Adja, B. Hammi, A. Serhrouchni, and S. Zeadally, "A blockchain-based certificate revocation management and status verification system," *Computers Security*, vol. 104, p. 102209, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016740482100033X

[14] Q. Aini, E. P. Harahap, N. P. L. Santoso, S. N. Sari, and P. A. Sunarya, "Blockchain based certificate verification system management," *APTISI Transactions on Management*, vol. 7, no. 3, p. 191–200, Nov. 2022. [Online]. Available: https://ijc.ilearning.co/index.php/ATM/article/view/1922

[15] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet x. 509 public key infrastructure certificate and certificate revocation list (crl) profile," Tech. Rep., 2008.

[16] P. C. Kocher, "On certificate revocation and validation," in *International conference on financial cryptography*. Springer, 1998, pp. 172–177.

[17] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X. 509 internet public key infrastructure online certificate status protocol-ocsp," Tech. Rep., 2013.

[18] L. Brünjes and M. J. Gabbay, "Utxo-vs account-based smart contract blockchain programming paradigms," in *Leveraging Applications of Formal Methods, Verification and Validation: Applications: 9th International Symposium on Leveraging Applications of Formal Methods, ISoLA 2020, Rhodes, Greece, October 20–30, 2020, Proceedings, Part III 9*. Springer, 2020, pp. 73–88.

[19] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," 2014. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[20] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf