Ethereum Attestation Service as a solution for the revocation of hardware-based password-less mechanisms

Biagio Boi
University of Salerno
Fisciano, Italy
bboi@unisa.it

Christian Esposito
University of Salerno
Fisciano, Italy
esposito@unisa.it

Jung Taek Seo
Gachon University
Seongnam-si, Republic of Korea
seojt@gachon.ac.kr

ABSTRACT

Hardware-based solutions are becoming more and more popular as a result of the increased need for practical and safe authentication methods. However, one of the key challenges in these systems is the lack of a robust mechanism to revoke compromised credentials effectively. The Ethereum Attestation Service (EAS), which uses the blockchain-based Ethereum platform to create a decentralized, tamper-resistant infrastructure for credential attestation and revocation, is presented in this article as a novel solution to this critical issue. By combining the transparency and immutability of blockchain technology with smart contracts and cryptographic techniques, the EAS enables secure and auditable management of certificates. The conducted study investigates the limitations of existing revocation methods of password-less mechanisms and proposes the EAS as a viable alternative. In the design phase, the paper demonstrates the system's efficiency in handling attestation requests, verifying attestations, and securely managing revocations. EAS excels in providing reliable revocation, thereby reducing the risks associated with compromised hardware-based passwordless systems. Moreover, this research explores the benefits of EAS-based revocation within the IoT context, where Physically Unclonable Functions (PUFs) face similar challenges as HSMs. Experimental results, obtained in a testnet environment, reveal reduced authentication times, making this solution suitable for real-time scenarios as well.

CCS CONCEPTS

• **Security and privacy** → *Authentication*; *Privacy-preserving protocols*;

KEYWORDS

Ethereum, Hardware-based Authentication, Password-less Mechanisms, Credential Revocation

ACM Reference Format:
Biagio Boi, Christian Esposito, and Jung Taek Seo. 2024. Ethereum Attestation Service as a solution for the revocation of hardware-based password-less mechanisms. In *The 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24), April 8–12, 2024, Avila, Spain.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3605098.3636004

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SAC '24, April 8–12, 2024, Avila, Spain
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0243-3/24/04.
https://doi.org/10.1145/3605098.3636004

1 INTRODUCTION

Data stored inside servers are increasing in an exponential way and with them also the possible threats. General Data Protection Regulation (GDPR) aims at giving users control over their data and establishing the way in which applications can access the data, but poor work has been done in trying to give users complete control over their credentials.

Decentralized authentication mechanisms, whose scope is to give users increased control over where their credentials are stored, are gaining increasing attention in the digital world as they offer a secure and efficient way to manage identity and access control. Despite the complexity of the architecture of such mechanisms, we're currently using these solutions for the authentication in most famous services; which incorporate them in a transparent and user-friendly way.

Single-Sing-On (SSO) defines the schema for decentralized authentication, where credentials move away from Service Provider (SP) and are stored in an Authentication Server (AS). Users do not have to remember and securely store a password for each SP, but have to store just one password, which is requested from the service leveraging on SSO. A widely known implementation of SSO is OpenID Connect (OIDC), different examples can be found in the literature; formal analysis of the security of such schema has been conducted by Fett et al. [3], who define the guidelines for the secure implementation. OIDC defines an AS, as responsible for storing and validating users' credentials; letting all privacy concerns open. The AS is the weakest link in such a schema since is responsible for the user authentication over multiple platforms, which in some cases adopts Multi-Factor Authentication (MFA) to prevent attacks. Anyway, Privacy is one of the major issues of this schema: an Identity Provider (IdP) is aware of its users' every sign-on attempt and the nature of visited websites. Similarly, any RP may learn users' identity information (e.g., email or social media account) when interacting with the IdP, even though this is not strictly necessary for authentication [13].

Password-less mechanisms are growing over the last few years giving as a means for the prevention of privacy issues that can come with adopting SSO-based solutions [8]. In such mechanisms, there is no need to manage passwords since almost all of them are based on a key pair where the private key is securely stored by users and the public one is stored directly by the SP. Hardware Security Modules (HSMs) represent the main technology used for implementing this approach since agree with key pair generation and secure storage of secret keys. Because of the nature of this system, no authority is responsible for verifying the authenticity of these credentials, making them valid just for assessing authentication but not for determining the validity of this authentication over time.

In the case of hardware-based implementation, usually, an expiration date is set, after which the credentials are no more usable by the holder. Such an approach for the revocation goes in contrast w.r.t. GDPR principles, where revocation typically should be effective within 24 hours after user reporting. The idea of our work is to propose a system for making hardware-based solutions valid also in terms of revocability on the basis of event, and not just on the basis of expiration data using a system based on attestation. An attestation is helpful for demonstrating that credentials are still valid or on the contrary, they are not valid anymore. Moreover, the revocation must be backward unlinkable, meaning that user anonymity must be guaranteed also after revocation.

Ethereum Attestation Service (EAS) is a new and growing technology, where multiple use cases have been formulated by creators but poor research has been done in trying to adopt it as a means for assessing the validity of hardware-based systems used for authentication. This paper discusses EAS architecture in order to exploit the main components and assess the influence that this technology may have on society. We show an additional use case of this service characterized by the revocation of Public Key Infrastructure (PKI) credentials with a simple implementation that leverages HSM for key management.

The main novelties introduced by our approach are related to the usage of attestation in assessing HSM validity. Such an attestation makes it possible to comply with GDPR revocation principles while removing a trusted authority. In particular, the adoption of EAS as a means for decentralized verification is an additional point that moves forward a fully decentralized authentication mechanism, where the users are the only responsible and the only owner of their credentials. Adoption of the presented mechanism can increase the use of password-less authentication by providing a reliable method for dealing with revocation. Such a method, as described in the following sections, can be easily used within the IoT context, where Physically Unclonable Functions (HSMs) can be considered a special case of HSM. The document is organized into six sections:

- The second section presents the related works on hardware-based password-less revocation systems, highlighting the drawbacks of existing solutions;
- The third section gives an overview of EAS, explaining the main components and its contribution to social good.
- In the fourth section we discuss how is possible to leverage EAS for handling with revocation of HSM, including a possible future development characterized by the IoT context;
- In the fifth section discusses the results of the proposed architecture, including privacy and security considerations;
- In the last section drawbacks and future possible developments are presented.

2 RELATED WORKS

Hardware Security Module (HSM) offers a reliable and strong system for creating password-less mechanisms; it is a tamper-proof device able to protect cryptographic processes, by managing the keys used for encrypting and decrypting the data. Such modules are able to sign documents using a mechanism based on Public Key Infrastructure (PKI). The identity cannot be spoofed since private keys are securely stored within the module.

One of the major investors in this context is FIDO (Fast IDentity Online) Alliance, which releases to users an HSM able to produce multiple PKI key pairs; the idea is to give services the public key and securely store the private one so that whenever the service want to assess the identity of the users can create a challenge based on the public key.

Despite research showing a good level of maturity of this technology, also in terms of usability [4] there is a big issue related to revocability [10]. In fact, as this is a decentralized authentication system, service servers do not need to contact the issuing party for checking the identity of users; but at the same time, this has some drawbacks. If the user loses access to his HSM, considering the case in which a thief stole his module, there is no possibility for him to recover credentials or worse, to deny access to the attacker.

Verhul [10] proposes a generic approach to the resolution of unlikable revocation, which can be applied to FIDO and other hardware-based security modules. Despite this work offering a good approach based on Certificate Revocation List (CTL) and on Online Certificate Status Protocol (OCSP); both have as drawbacks the centralization of Certification Authority (CA) making the password-less mechanisms dependent on a party. Such centralization, beyond going in contrast with the decentralization concept introduced by password-less authentication, can be dangerous considering that an attack on CA could mean the emission of certificates without any check, or in the worst case the publication of private keys associated with the emitted certificates.

With the advent of decentralization, solutions based on Distributed Public Key Infrastructure (DPKI) try to solve the problem of centralized authority. As discussed in the study conducted by [7], the major solutions based on DPKI are log-based PKI and Web of Trust, but problems exist for both solutions. The first one ignores data consistency in the log server, while the second one does not provide identity retention and is not friendly to new incoming members. De Filippi et al. [2] investigate Blockchain as an alternative to pure DPKI. They confirm that confidence in any blockchain-based system is achieved through a combination of multiple elements able to completely substitute the trust requested in a classical PKI model characterized by a CTL and OCSP. Thanks to this characteristic, the model of decentralized attestations starts to take place with a high level of transparency. Wang et al. [12] propose a schema where the CA-signed certificates and their revocation status information of an SSL/TLS web server are published by the subject as a transaction in the global certificate blockchain. In this way, the blockchain acts as append-only public logs to monitor CAs' certificate signing and revocation operations. 4.1.3 [12] discusses possible compromised CAs and publishing key pairs, showing how the miners are able to detect this behaviour and block the possible attacks in this direction. This approach can easily be extended to the case of password-less authentication mechanisms which leverage on HSM for the creation of key pairs. In this specif case the CAs can be substituted by the HSM itself, responsible for the releasing of new key pairs by signing a transaction on the blockchain.

Zhou et al. [14] propose a framework for verification, authorization, and recovery of a particular digital identity mechanism named Self-Sovereign Identity. Such an identity tries to decentralize the user identity but suffers the same problem of HSM where authority must be encharged of revoking the identities. They propose an

approach based on the Ethereum blockchain demonstrating how this blockchain can be used for multiple use cases.

The idea of having a certificate for each released key pair can leverage the support of the Ethereum blockchain, whose public registry makes verification by any party always possible. HSM, where revocation it is always been a big challenge can use this idea for solving this challenge and increase the usage of password-less authentication mechanisms. In the following section, EAS will be explored to highlight its potential within the context of attestation.

3 ETHEREUM ATTESTATION SERVICE (EAS)

Attestations can be helpful for verifying a contract, a fact, or more in general data. Over the years, the way in which these attestations are produced is radically changing, moving away from the physical word in favor of the digital one. One approach to making this digitalization possible is the adoption of digital signatures. These particular types of signatures are widely based on the PKI schema, where the sign is applied by computing the hash of the document and ciphering the content using the private key so that everyone can verify the signature using the public key. The digital signature needs the existence of a CAs responsible for the issuing of the signature and for the revocation of them. On the contrary, blockchain with its characteristics is paving the way to a decentralized way of handling digital signatures and existence in a given time.

Ethereum Attestation Service (EAS) leverages the Ethereum blockchain to create, manage and revoke the digital version of physical attestation. They allow anyone to create and validate attestations, EAS has the potential to transform how information is shared and verified online and throughout the Ethereum ecosystem. EAS is primarily based on two smart contract depicted in the Figure 1: one for registering a schema about any attestation topic, and the other for making attestations with that schema. These two smart contracts are published on the Ethereum blockchain and offer the necessary methods for the creation of the entire attestations life-cycle. An additional smart contract can be used for defining an optional schema resolver for more complex use cases.

In such a system three actors interact with attestations:

- *Attestors*: individuals or organizations interested in creating and signing attestations. They are responsible for adding the attestation to the Ethereum blockchain and making it available for verification. An attestor must be the owner of an Ethereum wallet; they are able to create any kind of attestation starting from a previously defined schema.
- *Verifiers*: individuals or organizations that want to assess the authenticity of attestations. They can perform this verification by checking the attestation on the Ethereum blockchain and checking that it has been signed by a trusted attestor. A verifier can be anyone with access to the Ethereum blockchain and the attestation's UID.
- *Users*: entities that use and rely on attestations to make decisions or take actions. They use the information provided in the attestation to verify the authenticity and integrity of the information being attested to, and they rely on the reputation and trustworthiness of the attestor as a form of backing for the attestation.

```
struct SchemaRecord {
    // A unique identifier of the schema.
    bytes32 uid

    // Optional schema resolver contract.
    address resolver

    // Whether the schema allows revocations explicitly.
    bool revocable

    // Custom specification of the schema.
    string schema
}
```

```
struct Attestation {
    // A unique identifier of the attestation.
    bytes32 uid

    // A unique identifier of the schema.
    bytes32 schema

    // The UID of the related attestation.
    bytes32 refUID

    // The time when the attestation was created.
    uint64 time

    // The time when the attestation expires.
    uint64 expirationTime

    // The time when the attestation was revoked.
    uint64 revocationTime

    // The recipient of the attestation.
    address recipient

    // The attester/sender of the attestation.
    address attester

    // Whether the attestation is revocable.
    bool revocable

    // Custom attestation data.
    bytes data
}
```

Figure 1: EAS Smart Contracts structure; on the left for registering a schema, on the right for making attestations with a given schema.

EAS serves as the base layer for coordinating, creating, and registering unique attestation schemas. This allows for interoperability and composability between various attestation protocols and solutions, enabling the attestation layer to evolve over time. EAS operates in two modes, on-chain, and off-chain: the first ones have directly stored in Ethereum blockchain, while the second ones are stored in decentralized storage like IPFS. Storing something in the on-chain version means that everyone can see the data and can access the information, while in the off-chain version, a hashing function guarantees a good level of privacy without any assurance that the attestation has been thoroughly reviewed and agreed upon by a decentralized network. A good compromise between these two types can be the publication in on-chain of off-chain UID, giving it in a timestamp and proving that data are not changed after that time.

The creation of attestation is subject to the validation of nodes that take part in consensus. Since EAS uses the Ethereum blockchain to make the transaction it is subject to the same process as Ethereum transactions.

3.1 EAS as a Public Good

EAS provides a decentralized, open-source solution for creating and verifying attestations leveraging one of the biggest public blockchains. In EAS there is no centralized authority or mechanisms based on tokenization, the only payment is related to the fees of the Ethereum network for registering the attestation. Moreover, the compromise between on and off-chain makes possible the availability of this service at a really low price which means that it is available for almost everyone in the network. This service includes multiple use cases that could have a big impact on society such as the attestation of property, the voting system, the product authenticity, the proof of possession, and so on. Finally, its last version includes support for files, making possible the direct link between attestation and file, which could be taken also as an opportunity for

the digital signature. Mechanisms for digital signatures nowadays are solving the previous problem related to usability but in some cases are still too weak; the adoption of attestation in this context can increase overall security.

4 ARCHITECTURE

In this section, we want to exploit EAS as a solution for managing the revocation of user-centric credentials released by HSM. HSM releases a new key pair (public and private key) every time the user needs new credentials. The Service Provider (SP) receives the public key as an identifier, while the user securely stores the private key in his HSM. Leveraging on signature schema it is possible to verify the identity of the user by exchanging a challenge when needed. To assess the validity of HSM and to prevent malicious node from stealing HSM and prevent unaware use of it, manufacturers usually leverages on centralized system for creating a revocation list containing the certificate associated with released HSM. When considering HSM in a decentralized context, where regulating authority does not exist, it is expensive to create a revocation list shared with the network and in some cases, the time used for the verification is too high.

The proposed system puts the user at the center of the authentication, in particular, he is the only one responsible for his credentials, removing from the loop every kind of CA. A decentralized environment offered by EAS, which leverages the Ethereum blockchain, will be the only one responsible for issuing and revoking the attestation. In the proposed system the *Attestator* and the *Verifier* can be considered as the same actor which can be any SP interested in implementing a revocable HSM-based authentication. The *Holder* can be considered as the owner of HSM and of a crypto wallet. At the beginning of the enrollment procedure, like in traditional HSM-based authentication, the user creates a new public key K_{pub} using his HSM. The public key K_{pub} is sent to the SP and associated with the user's account. The related private key K_{priv} will never leave the local device and will be stored in a secure memory managed by HSM itself. A this point the proposed architecture, resumed by sequence diagram in Figures 2,3 and 4, will take place and act as follows:

(1) *Attestation*: During the registration phase the *Attestator*, namely the SP, sends a challenge to the *Holder* asking him to sign it with the wallet private key WK_{priv}. The *Holder* sends an encrypted message, whose decryption using the wallet public key WK_{pub} must produce the address of the wallet. At this point, the *Attestator* produces an attestation A_h indicating as a recipient the *Holder* address and stores the generated UID_{A_h}. The Ethereum blockchain will be responsible for validating the transaction and sending the attestation to the indicated recipient. Such attestation will be securely stored within an Ethereum wallet which can be unlocked only using the private key WK_{priv} held by the user;
(2) *Verification*: At the authentication phase, the SP creates a challenge to verify the ownership of a previously registered HSM. The *Holder* sends the signed challenge back to the service, which verifies it with the stored public key K_{pub} and generates a request to the Ethereum blockchain for assessing the validity of the attestation A_h. If the K_{pub} and A_h are valid, then the *Holder* is authorized to log in to the system.
(3) *Revocation*: Whenever the *Holder* wants to revoke the credential used for registering on the SP system can perform a revocation request using his Wallet. In this case, the *Holder* sends a request to the Ethereum blockchain which will invalidate the attestation A_h so that every future verification performed by SP will generate a negative response, denying access to the system.

In such a system the SP is responsible for requesting the emission of attestation, in such a way the step described in (2) is valid only for the SP which is interested in implementing revocable HSM-based authentication. It is possible to extend the approach to all the pairs

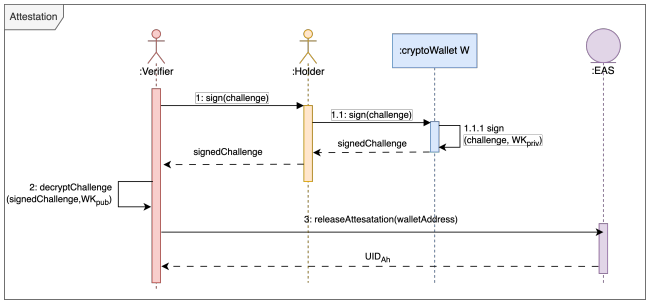

Figure 2: Sequence diagram of attestation

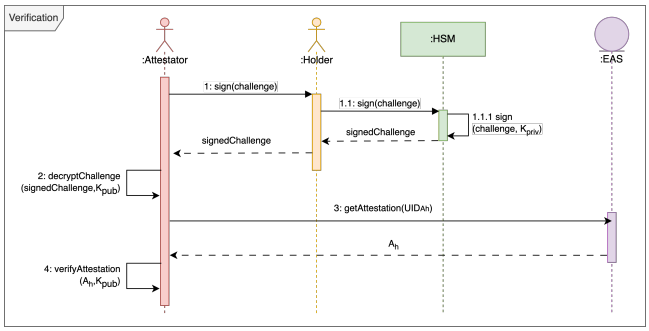

Figure 3: Sequence diagram of verification

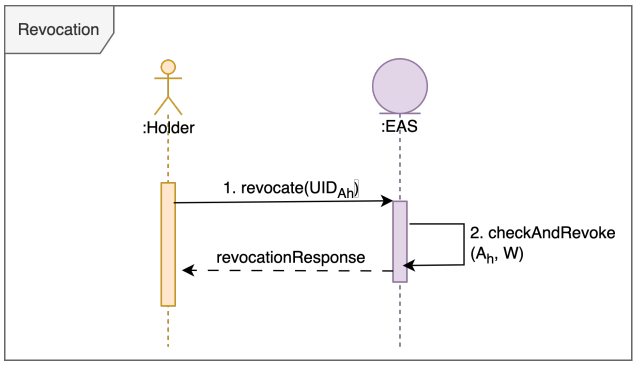

Figure 4: Sequence diagram of revocation

generated from one HSM by associating a certificate with the HSM identifier and checking for validity.

Acronym	Description
A_h	Attestation
UID_{A_h}	Identifier of A_h
K_{pub}	Public key used for registering on SP
K_{priv}	Private key associated with K_{pub}
W	Crypto wallet used for storing A_h
WK_{pub}	Public key associated with wallet W
WK_{priv}	Private key associated with WK_{pub}

Table 1: Acronyms used.

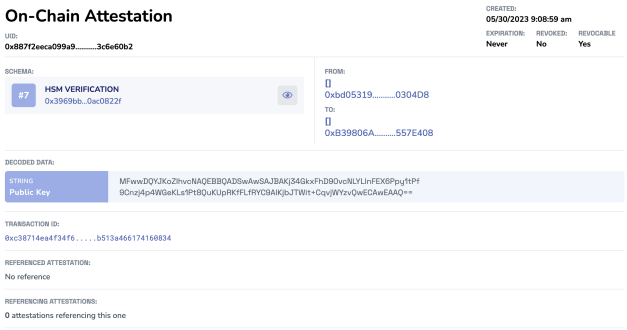

Figure 5: EAS Attestation for HSM verfication

4.1 Schema Definition

EAS requires the emission of a schema, a kind of template that must be used in the phase of attestations release. The smart contract depicted in 1 on the left agrees with the release of a new schema. The method to recall for the creation of a new schema is the register function: *register(string schema,address resolver,bool revocable)* which takes as parameters the schema structure, the address of smart contract resover if any and the revocability of the schema. Such a schema will be released once at the beginning and can be reused every time that SPs want to rely on it.

In our system, the schema must take into account the data used for the identification, namely the released public key, without any smart contract resolution and making possible the revocation. Notice that at this point the revocation is related to the schema itself, meaning that once the revocation has been declared, there is no possibility to release additional attestations.

The UID of the schema will be the nonce associated with the smart contract, in this way, it will be an increasing number. The released schema is publicly available so that everyone can recall the schema for attesting a new public key.

4.2 Attestation Release

The release of attestation is the phase in which the just released public key is published on the blockchain and associated with a particular Ethereum wallet.

In this case, differently from the schema, the operation leverages the smart contract in the right side of Figure 1 to the function *attest(tuple request)* where the tuple is characterized by field requested by attestation.

The user must connect himself to the service and send the data that want to demonstrate. Such data, as introduced in the schema definition is simply the public key released to the SP. The EAS is responsible for checking that in the phase of release, the owner of the request is the same as HSM. An additional server is put in place between the user and EAS where this check happens.

EAS offers the possibility to create both off and on-chain attestations, since we have just one attribute to demonstrate, the best choice is to generate an on-chain attestation. Such attestation is validated by the Ethereum network and is publicly available. Privacy concerns are not treated since the only exposed value is the public key and the link between this and a particular Ethereum wallet; which do not contain any sensitive information. As shown in Figure 5, the attestation contains the public key K_{pub} released by user HSM and transmitted to the SP. The attestation follows the schema *HSM VERIFICATION*, which has been previously defined.

4.3 Attestation Revocation

It is possible to handle with revocation of attestation by specifying this possibility at the issuing phase. More in particular, the revocation can happen for two motivation:

- *Expiration*: this is determined by the SP at the time of emission of the attestation. This could be useful to make invalid the attestation independently from the emission of an expressive revocation, particularly relevant for some sensible context. This is expressed by *Expiration* field in the Figure 5;
- *Revocation*: this happens when the SP or directly the user recalls the method to revoke a specific attestation. Notice that such a call cannot be done by anyone else and the revocability must be declared at the issuing phase. The status of revocation is expressed by *Revoked* in the Figure 5.

In case of an attack on a SP, all the credentials can be revoked, in such a way there is no possibility for malicious attackers to reuse them.

4.4 PUFs use case within IoT context

IoT devices, such as smart cards, payment terminals, and embedded systems, may integrate dedicated HSMs as secure elements. These secure elements provide a high level of protection for sensitive data, making them suitable for applications with stringent security requirements. The expanding usage of smart sensors within the automotive context is increasing the demand for security by default solutions, where HSM plays a vital role [6], also considering the challenges related to reliability. As a result, HSM currently existing within devices can be used for enhancing security [1] and creating secure authentication protocols with reduced power consumption, able to preserve the entire communication.

Physically Unclonable Functions (PUFs) can be considered as a subcategory of HSM and can be utilized as a possible hardware solution for identification and authentication in an IoT context. Considering that PUFs extract unique hardware characteristics, they can be considered as a definitive solution for secure key generation [9]. The main drawbacks of such an approach are related to the stability of the produced key and to the revocation. The stability concerns are caused by the randomness of transistors, but numerous studies

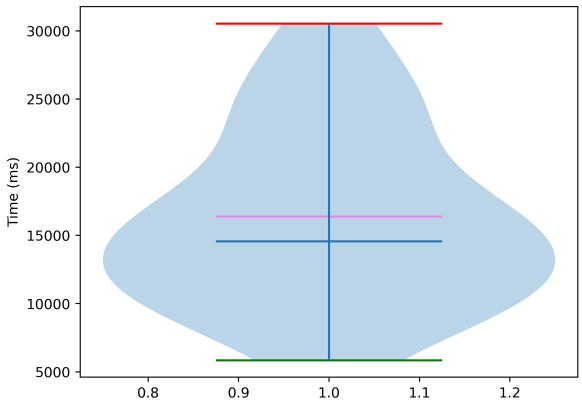

Figure 6: Time elapsed for attestation

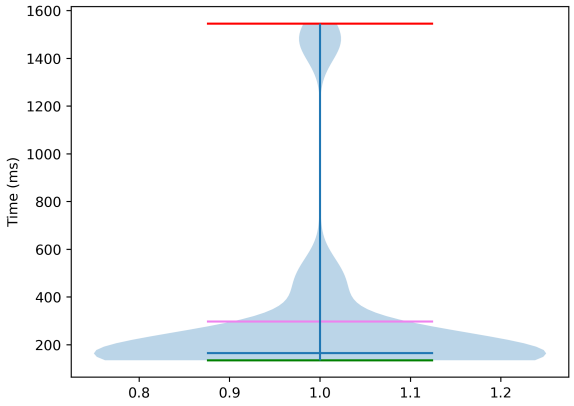

Figure 7: Time elapsed for verification

have been proposed in the literature to solve this problem with the usage of Error Correction Code (ECC) [5] [11]. The problem of revocation, instead is similar to the case of traditional HSM, where no mechanism for invalidate the credentials exist. Considering the case of a decentralized approach in which a device is tampered or it is has been stolen, there is no chance to deny access to that specific device. The proposed approach can be extended to the IoT context, by proposing a mechanism for securing hardware-based credentials in a decentralized context. The Holder as considered in Figures 2, 3 and 4 can be considered as the stakeholder interested in securing the IoT devices. In such a solution, the fully decentralized approach can be applied, where no password or centralized server responsible for the authentication is deployed.

5 RESULTS

The proposed schema has multiple advantages in terms of privacy and security. An attacker interested in stealing the HSM will not be able to use it thanks to the certificate associated with it if the owner revokes the certification on the EAS.

In this section, we are interested in assessing the performance of the proposed method from two perspectives: performance and security.

5.1 Performance

To perform evaluation we tested the EAS in the Sepolia Testnet environment, which is an Ethereum-based test environment with the same capability as the real environment. An iMac with a 3,3 GHz Intel Core i5 6 core processor and 16 GB 2667 MHz DDR4 RAM has been used to perform the evaluation. The operations subject of evaluation are the same as discussed in the architecture proposal and are attestation, verification, and revocation.

5.1.1 Attestation. In the attestation phase, a request for a transaction is sent to the blockchain. Such transactions must be validated by nodes before being considered valid. As shown in Figure 6, the highest time for performing the request requires 16.38s to be completed. This operation is performed whenever a new user wants to join the SP server and includes of exchange of public key and the creation of attestation. As it is possible to notice, despite some requests may require up to 30s to be completed, this is only a bound because most of them are performed in a medium time. Despite the fact that this may appear like a very long period for performing attestation, two considerations must be taken into account. First off, the time can be cut down by raising the transaction's gas limit; second, the duration is acceptable because the attestation process only needs to be completed once for each new user.

5.1.2 Verification. In the verification phase, there is no transaction with blockchain but the only operation needed is to check the validity of proposed attestation. As shown in Figure 7, verification can be performed in a mean time of 297.62ms. This phase is particularly important because the SP will always need to do a validity check on an attestation during the authentication phase. The maximum time per operation is achieved only when provider APIs are full of requests and correspond to 1.54s.

5.1.3 Revocation. During the revocation phase, an operation on the blockchain must be performed. In particular, the event of Revocation must be announced from nodes after a request from the Holder. The average time for the revocation is really similar to that used for the attestation, as shown in Figure 8, with a mean of 16.02s. Anyway, considering that revocation will be issued only when the holder loses ownership over HSM, it is an acceptable time.

5.2 Privacy and Security

Considering the HSM as a decentralized password-less mechanism able to authenticate a user without sharing additional details about his identity, it is important to guarantee the same level of privacy with the addition of the attestation system. As described in the system architecture section, the released attestation will contain only the hashed value of the device identifier, making it impossible for the attacker to perform an indexing based on such value. Such an identifier is unique by design and belongs to the company that released the certificate.

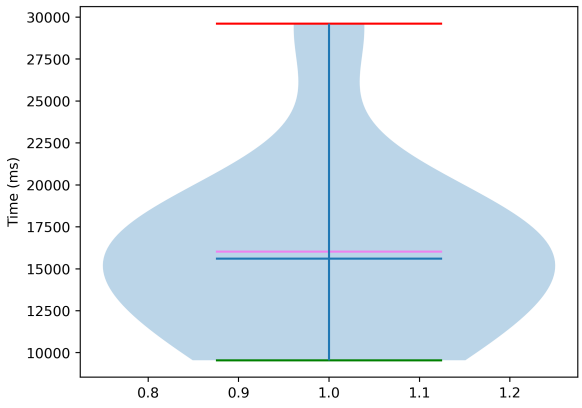

Figure 8: Time elapsed for revocation

Even if the user's wallet identifier is exposed in the blockchain, it is impossible for the attacker to use such a wallet without the physical presence of HSM. Complementary, if the attacker finds an HSM and wants to use it, will not be possible since the user will be able to revoke the certification by accessing his wallet.

6 CONCLUSION

The increasing reliance on password-less mechanisms for secure authentication in various domains has raised concerns regarding their vulnerability to compromise and the lack of efficient revocation mechanisms. This paper examined the potential of the Ethereum Attestation Service (EAS) as a solution for addressing the revocation challenges associated with hardware-based password-less mechanisms.

By leveraging the transparency and decentralized nature of the Ethereum blockchain, EAS enables the secure attestation and revocation of hardware-based password-less credentials. Moreover, EAS provides a scalable and efficient revocation mechanism, enabling the timely and widespread dissemination of revoked credentials to the parties. This is achieved through the use of smart contracts and decentralized consensus mechanisms, which automate the revocation process and eliminate the need for centralized authorities.

While EAS shows great potential, it is important to acknowledge some limitations and challenges. These include the fee which is necessary to pay for the registration of attestation, whose hybrid approach consisting of mixed on-chain and off-chain registration is not fully solving the problem and is limiting the power of this attestation.

Future development of this schema can involve the IoT context and PUFs, where the key is generated by leveraging hardware characteristics of devices, by completely removing the usage of passwords as a means for authentication. Future research in this area holds tremendous promise for revolutionizing how device identities are established and managed in the modern era. We have planned to conduct an experimental evaluation in order to better understand the impact of this solution on power-constrained devices.

ACKNOWLEDGEMENTS

This work was partially supported by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

REFERENCES

[1] Antonio J Cabrera-Gutiérrez, Encarnación Castillo, Antonio Escobar-Molero, José A Álvarez-Bermejo, Diego P Morales, and Luis Parrilla. 2022. Integration of hardware security modules and permissioned blockchain in industrial iot networks. *IEEE Access* 10 (2022), 114331–114345.

[2] Primavera De Filippi, Morshed Mannan, and Wessel Reijers. 2020. Blockchain as a confidence machine: The problem of trust & challenges of governance. *Technology in Society* 62 (Aug 2020), 101284. https://doi.org/10.1016/j.techsoc.2020.101284

[3] Daniel Fett, Ralf Küsters, and Guido Schmitz. 2017. The web sso standard openid connect: In-depth formal security analysis and security guidelines. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, 189–202.

[4] Sanam Ghorbani Lyastani, Michael Schilling, Michaela Neumayr, Michael Backes, and Sven Bugiel. 2020. Is FIDO2 the Kingslayer of User Authentication? A Comparative Usability Study of FIDO2 Passwordless Authentication. In *2020 IEEE Symposium on Security and Privacy (SP)*. 268–285. https://doi.org/10.1109/SP40000.2020.00047

[5] Ashwija Reddy Korenda, Fatemeh Afghah, Bertrand Cambou, and Christopher Philabaum. 2019. A proof of concept SRAM-based physically unclonable function (PUF) key generation mechanism for IoT devices. In *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 1–8.

[6] Carson Labrado and Himanshu Thapliyal. 2019. Hardware security primitives for vehicles. *IEEE Consumer Electronics Magazine* 8, 6 (2019), 99–103.

[7] Yannan Li, Yong Yu, Chunwei Lou, Nadra Guizani, and Lianhai Wang. 2020. Decentralized public key infrastructures atop blockchain. *IEEE Network* 34, 6 (2020), 133–139.

[8] Viral Parmar, Harshal A. Sanghvi, Riki H Patel, and Abhijit S. Pandya. 2022. A Comprehensive Study on Passwordless Authentication. In *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*. 1266–1275. https://doi.org/10.1109/ICSCDS53736.2022.9760934

[9] Alireza Shamsoshoara, Ashwija Korenda, Fatemeh Afghah, and Sherali Zeadally. 2020. A survey on physical unclonable function (PUF)-based security solutions for Internet of Things. *Computer Networks* 183 (2020), 107593.

[10] Eric R. Verheul. 2016. Practical backward unlinkable revocation in FIDO, German e-ID, Idemix and U-Prove. Cryptology ePrint Archive, Paper 2016/217. https://eprint.iacr.org/2016/217 https://eprint.iacr.org/2016/217.

[11] Rui Wang, Georgios Selimis, Roel Maes, and Sven Goossens. 2020. Long-term continuous assessment of SRAM PUF and source of random numbers. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 7–12.

[12] Ze Wang, Jingqiang Lin, Quanwei Cai, Qiongxiao Wang, Daren Zha, and Jiwu Jing. 2022. Blockchain-Based Certificate Transparency and Revocation Transparency. *IEEE Transactions on Dependable and Secure Computing* 19, 1 (Jan 2022), 681–697. https://doi.org/10.1109/TDSC.2020.2983022

[13] Zhiyi Zhang, Michał Król, Alberto Sonnino, Lixia Zhang, and Etienne Rivière. 2020. El passo: privacy-preserving, asynchronous single sign-on. *arXiv preprint arXiv:2002.10289* (2020).

[14] Tong Zhou, Xiaofeng Li, and He Zhao. 2019. EverSSDI: blockchain-based framework for verification, authorisation and recovery of self-sovereign identity using smart contracts. *International Journal of Computer Applications in Technology* 60, 3 (2019), 281. https://doi.org/10.1504/IJCAT.2019.100300