A review of Ethereum-Blockchain-Based Technology of Decentralized Smart Contract Certificate System

Merlijn Mulder
1009532
ISTD
Singapore University of Technology and Design (SUTD), Singapore
merlijn_mulder@mymail.sutd.edu.sg

Abstract

Digital certificates in the current system are prone to tampering, cyberattacks or unwanted voiding. These liabilities can be resolved in a blockchain implementation using smart contracts due to their transparency and immutability. The paper reviews such a proposal called Open Certificates. It harnesses smart contracts within a 3-layer authority system taking care of the 3 specified roles of a certificate system. Whilst it is found that such a system can be highly effective in resolving the described issues and provide an efficient way of validating, the paper does pose some vulnerabilities. This includes the elections of the highest authority (the system managers). Furthermore, as only states of the certificates are stored in different smart contracts a user will be unable to actually download a PDF an present it as proof to a less modern company/institute that may require it. In this paper review, I propose a solution to these vulnerabilities harnessing two other proposed systems which could potentially be integrated in the idea of open certificates.

Keywords:blockchain, Ethereum, smart contracts, Certificate Systems, security, efficiency

Introduction

Due to the development in the IT industry, around the world, we are moving from paper certificates (which are difficult to preserve, verify, and anti-counterfeit) to a digital version. Whilst this reduces the cost of issuance and inconvenience in management and preservation, it proves to present other liabilities. Firstly, information is stored in a database providing applications and interfaces by (third-party) agencies. This invites digital attacks and tempering, effectively voiding certificates, by malicious adversaries. Secondly, in case the agency in question disappears the data in their databases containing the valid certificates is possibly voided and loses accessibility. Thirdly, verifying certificates is inefficient due to a variety of custom tools provided by agencies that are not industry standards. This variety leads to the difficulty of verifying the correct honest software and tools are provided. As a way to counter these problems within this technological development is blockchain. This technology embodies distributed data storage, point-to-point transmission, consensus and cryptographic algorithms. Within this decentralized combination of techniques, it tries to ensure anti-tampering and traceability. With the introduction of smart contracts to the concept, blockchain has the potential to host a decentralized certification system that is easy to verify and ensures anti-tampering due to its immutable design and anti-counterfeeling due to its transparency. Therefore Xie et al. (2020) proposes such an implementation to share certificates whilst enabling trust among all involved parties. Within this paper, we will quickly delve into the techniques behind this solution and discuss potential shortcomings and unclarities of the authenticator structure and the data storage on the chain.

Main idea and proposed system

The paper specifies three roles of certificate systems:

1. Roles of designing, issuing and managing certificates.
2. Roles of certificate acquisition and acceptance.
3. Roles in need of verifying certificates, which can be further categorized into four types, thus requiring different functions of the system.
   (a) Responsible for a design according to the real situation in granting the certificate
   (b) Administrative functions, authorizing certificate issuing. This tracks system states and eliminates chaos.
   (c) Certificate acquisition and acceptance to the person in question.
   (d) Verification of a certificate.

Feasible and authentic administration is needed to provide complete information and prevent dishonest certificates to verify and manage the process. It should be independent and reliable to function adequately. Existing solutions like BlockCert solve certificate issuing and verifying related problems using Blockchain and LLP even solves data storage problems using a decentralized management system on top of that. However, they don't solve all potential difficulties and problems. A decentralized certificate system based on Blockchain and smart contracts might allow trustful sharing between receivers and the corresponding authority.

The protocol proposed for this system, *Open Certificates*, standardizes certificate information to provide multi-system compatibility. Within this protocol, only the data essential to the certificate is stored according to the following categories: *Issuing Agency data (certificate authorities), Issuer Data (issuer information and public key), Template Data(shared certificate details), Reciever Data (name and hashed identity for privacy) and Certificate Data (Template and receiver information are linked with issuance details).* As all the data is stored in smart contracts, information access can only be achieved through the *CertManage Contract*, checking the

caller's permissions against their address. Therefore only authorized users can add templates and issue and revoke certificates. The information is stored in the following contracts: *CertManage (Acces permissions), CertAgency (issuing authority data), Certifier (issuer details), CertTemplate (certificate templates), Reciever (recipient information), Certificates (Manage issued certificates)*. The specific role of the Certificate Template is optimizing data storage as enables bulk issuance of identical certificates which reduces redundancy within the chain. The public key is generated using the elliptic curve ecdsa-secp256k1 algorithm.

The proposed concept adopts a smart contract to realize authority management and certificate issuing, verification and revocation handles authority management in a three-level adoptable mode:

1. *System Managers*
   (a) Oversees the certificate-issuing agencies by approving and registering them after verifying their qualifications
   (b) Manages the information through *CertAgency and CertManage contracts*.
   (c) Can be selected by a committee or designated by a smart contract deployer
2. *Certificate Issuing Agencies*
   (a) Handling the authority of certificate issuers by adding their details and public keys to the *Certifier Contract* using their private key.
   (b) Registering is done by submitting its public key and information to a system manager for approval.
3. *Certificate Issuers*
   (a) Issues the certificate to the user by adding or revoking their certificate whilst verified with the *CertManage contract*.
   (b) Registered in the Certifier contract and authorized by an issuing agency.

The non-three-level alternative removes the system managers from the hierarchy allowing the agencies to act independently. Then the agencies can register themselves and authorize without a systems manager's intervention. The data itself is stored in a smart contract releasing the relevant data when issued, verified and revoked. Access will be checked by authorities and accessing is only possible by calling the smart certificate management contract. This removes the need for third parties as the entire system relies on smart contracts within the Ethereum-based blockchain. As the system exports data compatible with the Open-badges protocol found in BlockCert, it can conveniently be integrated into existing solutions in a user-friendly manner.

certificate issuance

1. Issueres call CertaManage to add a certificate template in CertTemplate contract.
2. An issuer selects a template from the contract, gathers the receiver information and signs the data
3. Permissions are verified via CertManage and issuing authority is checked by CertAgency and Certifier Contract.
4. In the case of a new receiver, their information is added to the receiver contract
5. Certificate-related data is put in the Certifier contract and the issuer informs the receiver.
6. A copy is sent to the receiver containing the number, hash, data, etc.
7. A receiver can share this data for verification if needed with other parties.

Revocation is done by the issuer or agency submitting a certificate hash to the CertManage contract. When permissions are verified then revocation records are added to the Certifier and CertAgency contract. Expiration is validated during verification and does not automatically trigger revocation.

Verification process

When a verifier wants to verify a certificate, this can be done according to the process in Figure 1. Note that access to all information is done via the CertManage contract. The process in the figure can be described in the following 7 steps:

1. The smart contract calls the blockchain to make sure the certificates exists on the chain and extract certificate information.
2. A signature is calculated which needs to match with the certificate found on the blockchain.
3. It is checked whether the public key matching with the signature aligns with the keys found in the template and the key pair issued by the verifier.
4. Verify that the issuer has not revoked the certificate.
5. Confirm that the issuer is active and allowed to issue certificates.
6. Check the expiry date on the certificate to ensure its validity at the time of verification.
7. Return all results to the smart contract and the verifier calling the contract.

Observations and Analysis

As can be observed in the proposed system explained above, both three-level authority management and the alternative non-three-level authority management are proposed, depending on the situation.

It argues that the system is fully decentralized as it doesn't rely on third parties and the system stores all the information solving the liability problems other solutions present.

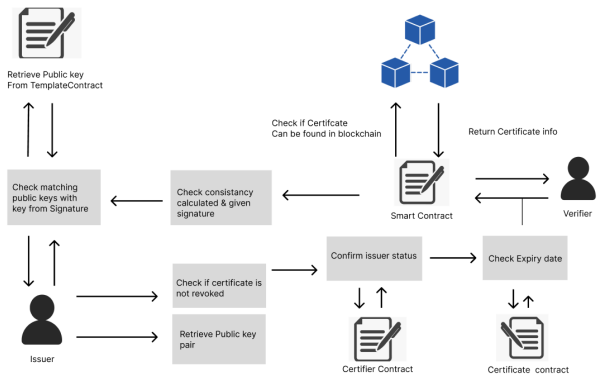

Figure 1: Verification Process

Results

This leads to the comparison between other (Blockchain) solutions found in figure 2, where the Xie et al. claims to solve all the shortcomings that present liabilities for the system mentioned earlier. It highlights that *Open Certificates* is an independent system whilst the other solutions are not and it is completely decentralized according to their reasoning. Furthermore, it claims it is the only solution in the comparison that holds the full certificate information. However, where it might hold all the necessary information for validation, it preserves states in different contracts and an old-fashioned PDF or paper-like version cannot easily be retrieved.

	BlockCert	LLP	SPROOF	This System
Smart contract	No	Partial	No	Yes
Permission management	No	Yes	No	Yes
Independency	No	No	No	Yes
Storage	Third party	Third party	Third party	Self-contained
Privacy protection	Yes	Yes	Yes	Yes
Full certificate information	No	No	No	Yes
Decentralization	No	No	Yes	Yes
Transparency	Yes	Yes	Yes	Yes
Completeness	No	No	Yes	Yes

Figure 2: Comparison between other solutions Xie et al. (2020)

Pros and Cons (Lessons learned)

As discussed and seen in the results the proposed solution provides more advantages over other solutions that can be essential to a functioning certificate system. Furthermore, it solves the liabilities of more conventional digital certificate systems, as stated earlier in the paper as it harnesses Blockchain. This means that using such a solution in the future will make the validation process more efficient and the system will become more reliable due to its transparency and immutability. That is as long as the contracts contain all three roles such a system should execute. What the paper fails to mention is how adding a contract later on can work in the

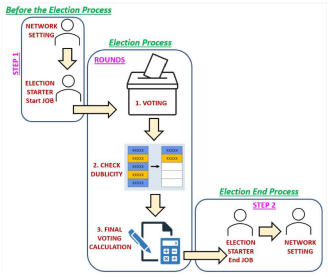

Figure 3: A secure sped up blockchain election system based on PoS by Naz et al., 2024

case the system is in use but needs to be revised or slightly adjusted. Will the system with this hierarchy still be able to revoke or validate certificates held in older contracts before a hard fork? The paper does not seem to discuss the possibility of the system being flexible if needed. Assumed is a correct implementation that should last to provide the most secure system. Information leakage of a user's privacy is prevented as receiver information is not fully preserved. This means that within the three-level authority management system, the system will be affected if the private key is lost or compromised as some contracts cannot be validated or adjusted anymore. The paper proposes a voting schema in authority management to make decisions according to the other's opinions to avoid the effect of the lost key. This will give system managers the right to affect the chain. As said a system manager oversees the agencies and is selected by a committee. This means that it would be a central authority within the system combining two approaches. That is if this authority would be a said organisation. This is not made very clear in the paper. If this is done in the way PoS chooses its validators, in a manner that is somewhat random, offering a proper incentive whilst building the chain one can argue that this authority is decentralized as well. An election process, where the issuing authorities show their vote for every x amount of time assures that if there are faulty managers in the system their malicious attempts can be countered. A way to ensure a fair voting and selection process is by randomly devising all system managers into different pools and selecting one from each pool. Then the issuers can cast their vote, the system checks for duplicates, and based on a weighted draw the new manager for the next x amount of time is selected and will be rewarded for their work if the newly issued certificates end up on the longest chain. Such a process can make the process more efficient with a large number of system managers as described by Naz et al. (2024) and seen in figure 3.

Furthermore, the systems only use states of certificates to prevent forgery and optimize the data storage. This means the certificate itself cannot be found in the chain nor does it revoke the certificate automatically when it expires. It only handles small pieces of data like the authentication information which is more memory efficient. Tellew and Kuo (2022) proposes a different solution for storing full training certifi-

cates: Certificate chain. As there is a limitation for usual transaction sizes, it implements a dividing-and-merging algorithm, splitting each certificate into 30 kb slices which are reconstructed into a byte array in the chain. This doesn't limit the smart contract to PDF files but can also incorporate other files. The results are shown in figure 4, and we can conclude that a blockchain system using slicing techniques can efficiently carry full PDF certificates. This is different from the system described in this paper only using states in smart contract, not storing the entire PDF. Whilst it does deploy options for efficient verifications and issuing of the certificates it does not store the entire certificate-like certificate chain. As some agencies might prefer also storing entire certificates, which the system is currently lacking, a merge with a solution like CertificateChain might present a more complete solution. As the Open Certificates do include a template contract, this can contain slices of recurring data in all certificates whilst the rest of the data is stored in a byte array in the certificate contract. The entire pdf can then be resembled from this byte array including pointers to the template contract where needed that holds the recurring information found in the pdf. Therefore an actual pdf of the certificate can be collected from the blockchain.

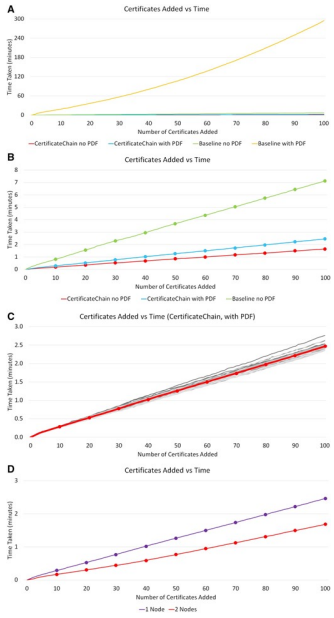

Figure 4: Baseline models vs model carrying pdf using slicing Tellew and Kuo (2022)

Furthermore using smart contracts the opposed solution makes certificate management and issuing more secure and independent. Users can access and send the data in a confidential way as the chain is immutable and verification is done through a multi-layer authentication system using private addresses and keys, verifying the authenticity of the issued certificate. It harnesses the key aspects of blockchain,

Conclusion and Future Direction

As specified earlier a certificate system has three roles. The proposed Open Certificate protocol manages this via different smart contracts within the Ethereum-based blockchain. These smart contracts are subdivided into responsibilities of these roles which can be accessed via different authorities in the chain. These being *system managers*, *Issuing Agencies*, *Issuers*. An alternative is also discussed removing the need for *system managers*, and therefore not having a more central checking authority within the system. Whilst this is not specified in the paper this role can be further decentralized by a PoS-based voting system, allowing different managers on the chain, lowering risks of implications of a malicious manager on the chain if anybody could take up said functions. Furthermore whilst some information is stored in the chain, the entire PDF holding a certificate is not. This means that an issued certificate does not look like what we are used to but is comprised of different smart contracts. To add this functionality, characteristics from a smart slicer solution like a Certificate chain could be implemented. This ensures certificates can also be used outside of the chain if it is not accepted at a specific place. This would be a future direction for one of its shortcomings. However, as described in the paper, the next step would be to extend the smart contract by adopting a new authority management schema. This can lead to the development of more general processes defining the protocol for storage of digital data in smart contracts for general purposes.

References

Naz, S., Siddiqui, M. J., & Lee, S. U.-J. (2024). S&sem: A secure and speed-up election mechanism for pos-based blockchain network. *Mathematics*, *12*(20), 3263. https://doi.org/10.3390/math12203263

Tellew, J., & Kuo, T.-T. (2022). CertificateChain: decentralized healthcare training certificate management system using blockchain and smart contracts. *JAMIA Open*, *5*(1), ooac019. https://doi.org/10.1093/jamiaopen/ooac019

Xie, R., Wang, Y., Tan, M., Zhu, W., Yang, Z., Wu, J., & Jeon, G. (2020). Ethereum-blockchain-based technology of decentralized smart contract certificate system. *IEEE Internet of Things Magazine*, *3*(2), 44–50. https://doi.org/10.1109/IOTM.0001.1900094