# Poster: Transparent Certificate Revocation for CBE Based on Blockchain

Qin Wang*[3], Rujia Li*[1,2], Qi Wang[1], and David Galindo[2]

[1]Southern University of Science and Technology, Shenzhen, 518055, Guangdong, China
[2]University of Birmingham, Edgbaston, B15 2TT, Birmingham, United Kingdom
[3]Swinburne University of Technology, Melbourne, VIC 3122, Australia

***Abstract*—In the certificate-based encryption (CBE) scheme, an illegal certificate revocation may disable the decryption capabilities of the corresponding certificate's owner. In this work, we propose a transparent certificate revocation mechanism for the certificate-based encryption, where a smart contract is involved as an agent to assist the certificate authority in managing the revocation. Our solution provides a transparent revocation procedure with incentives for honest actions. The preliminary analysis shows that our scheme is feasible and secure.**

***Index Terms*—Certificate Revocation, Certificate-based Encryption, Smart Contract.**

## I. Introduction

Certificate-based encryption (CBE), firstly introduced by Gentry [1], has received considerable attention [2] [3]. CBE is an intermediate paradigm that retains the desirable properties of public-key cryptography and identity-based encryption. In particular, it mitigates the certificate revocation problem, where the revocation is achieved by stopping the issuance of an implicit certificate for the revoked public key. In a CBE model, an up-to-date certificate must be obtained from Certificate Authority (CA) since it is used as a partial decryption key.

However, such a mechanism relying heavily on CA presents several concerns. (1) CA may arbitrarily revoke a valid certificate and repudiate her actions, and then indirectly making decryption fail. For example, Alice sends a ciphertext to Bob, but the evil CA has already revoked Bob's certificate without his permission. There is no way for Bob to obtain an up-to-date certificate and, as a result, he cannot decrypt the ciphertext; (2) Users cannot blame CA due to the absence of valid evidence on her malicious behaviour; (3) There is a lack of incentive for CA to behave honestly.

Several blockchain-based revocation schemes have been proposed (*e.g.,* [4]). However, they focus on the transparent revocation of traditional PKI schemes. To the best of our knowledge, this work proposes the first smart contract-based certificate revocation solution designed for CBE.

## II. Our Approach

Our scheme utilizes a smart contract as a transparent agent to manage the revocations. The user is required to send revocation requests to the smart contract. Then the smart contract checks the validity of the requests, including the authenticity of identity, the expiry date of the certificate, etc. Next, the smart contract periodically transfers the valid requests to CA. Finally, CA releases the new *reconfirmation* status (stopping the issuance of certificates for the revoked public key). Our solution focuses on the certificate revocation procedure. Here, we provide a generic CBE construction and then emphasize the enhancement of the certificate revocation algorithm.

### A. Generic CBE Construction

- **Key Generate** $(msk, pms) \leftarrow \mathsf{Gen}(1^\lambda, n)$. The algorithm inputs a security parameter $\lambda$ and (optionally) the total number of time periods $n$. It returns the certifier's master secret $msk$ and public parameters $pms$ that include master public key $mpk$.
- **Set Key**. $(PK, SK) \leftarrow \mathsf{Set}(1^\lambda)$. This algorithm is run by the user and outputs the user's key pair $(PK, SK)$.
- **Certificate**$^\star$ $Cert_i \leftarrow \mathsf{Cert}(msk, i, user, PK)$. At the start of each time period $i$, CA inputs certifier's master secret $msk$, user's information $user$ and public key $PK$, and outputs the certificate $Cert_i$.
- **Encrypt** $ct \leftarrow \mathsf{Enc}(m, i, user, PK)$. The algorithm inputs $(m, user, PK)$ at time period $i$, and returns a ciphertext $ct$ on message $m$.
- **Decrypt** $m/\perp \leftarrow \mathsf{Dec}(Cert_i, SK, ct)$. The algorithm inputs $(Cert_i, SK, ct)$ at time period $i$, and then returns a message $m$ or the special symbol $\perp$ indicating a decryption failure.

### B. Combination with Blockchain

We decouple the certificate updating algorithm of CA, and define several operations managed by the smart contract. In our scheme, three entities are involved (see Fig.1): *user*, *CA* and *smart contract*. The user (representing both sender and receiver) runs either the encryption and decryption algorithm. CA creates the smart contract (SC) and updates the final certificate status. SC bridges users and CA, managing the revocation, including (1) recording the revocation conditions; (2) setting the incentive policies; (3) verifying the eligibility of users' revocation requests according to predefined policies. The detailed steps are described as follows.

* These authors contributed equally to the work.

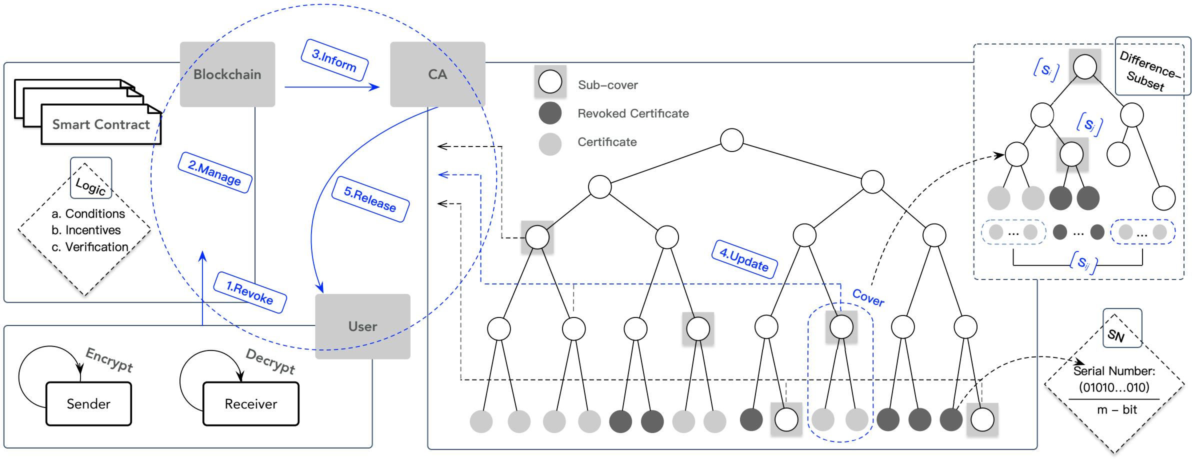


Fig. 1. Blockchain-based CBE Model

*Revoke*. The user sends the revocation requests to the SC.

*Manage*. SC manages requests using the following logic: 1) stores the revocation conditions such as the expiry date; 2) sets the incentives policies for CA to motivate her honest actions; 3) verifies the eligibility of users' revocations according to the predefined policies.

*Inform*. The SC informs [5] CA that the revocation requests are ready, and sends the approvals to CA.

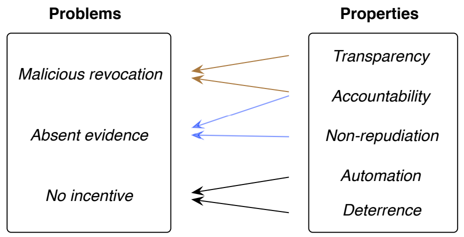
*Update*. CA updates certificates' status through a binary tree. CA arranges at most $2^m$ clients as leaves in a $m$-level binary tree. Each client is embedded by a unique $m$-bit serial number (SN) in its leaf nodes, and SN provides both identities and positions in the tree. The revocation is represented by the deletion of a leaf's sub-cover nodes (see Fig.1). In the meanwhile, to improve the efficiency of updating, the difference sub-cover approach [6] is adopted. Note that $S_{ij}$ denotes the set of leaves in the subset of $S_i$ but not in $S_j$.

*Release*. The updated result, denoted as *reconfirmation* certificate $Cert_i$, is sent to users for their decryption.

## III. Discussion

The combination of smart contracts and CBE brings us the following properties. *Transparency*: revocation data and related conditions are transparent to the public. *Accountability*: the revocation requests are globally auditable and accountable. *Non-repudiation*: CA cannot deny her illegal revocation due to transaction-based evidence. *Automation*: revocation operations are automatically executed. *Deterrence*: CA with illegal revocations will be punished.

Now, we show how these properties mitigate problems on malicious revocation, absent evidence and poor incentive. Firstly, a centralized execution easily breeds malicious revocations. The properties of transparency and accountability make the procedure publicly visible and auditable. Secondly, the smart contract receives revocation requests from users and then pushes valid ones to CA through transactions, where these transactions are used as evidence to detect illegal revocations. Thirdly, our scheme automatically provides cryptocurrency-based rewards/punishments under the predefined policies in SC for the CA's actions, which motivates the CA to behave honestly.



## IV. Summary

In a traditional CBE scheme a CA may maliciously revoke certificates and deny her involvement, hence preventing users from decrypting. We have presented a transparent certificate revocation mechanism for CBE, which employs the smart contract as an agent to prevent malicious revocation. We also note that our approach, which directly reveals the revocation identity to a smart contract, may violate users' privacy. Meanwhile, it may face the same scalability issues inherited from the blockchain. We leave them as potential future work.

## Acknowledgement

R. Li and Q. Wang were supported in part by the Ministry of Science and Technology (MOST) of China under Grant No. 2017YFB1003102.

## References

[1] C. Gentry, "Certificate-based encryption and the certificate revocation problem," in *Crypto*. Springer, 2003, pp. 272–293.
[2] D. Galindo, *et al.*, "Improved certificate-based encryption in the standard model," in *JSS*, vol. 81, no. 7. Elsevier, 2008, pp. 1218–1226.
[3] J. K. Liu and J. Zhou, "Efficient certificate-based encryption in the standard model," in *SECCRYPT*. Springer, 2008, pp. 144–155.
[4] Z. Wang, *et al.*, "Blockchain-based certificate transparency and revocation transparency," in *FC'18*. Springer, 2018, pp. 144–162.
[5] Oralize. [Online]. Available: http://www.oraclize.it/
[6] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Crypto*. Springer, 2001, pp. 41–62.