A Note on "CABC: A Cross-Domain Authentication Method Combining Blockchain with Certificateless Signature for IIoT"

Zhengjun Cao, Lihua Liu

Abstract. We show that the authentication method [Future Gener. Comput. Syst. 158: 516-529 (2024)] cannot be practically implemented, because the signature scheme is insecure against certificateless public key replacement forgery attack. The explicit dependency between the certificateless public key and secret key is not properly used to construct some intractable problems, such as Elliptic Curve Discrete Logarithm (ECDL). An adversary can find an efficient signing algorithm functionally equivalent to the valid signing algorithm. We also correct some typos in the original presentation.

Keywords: Certificateless signature, signing algorithm, verification algorithm, forgery attack

1 Introduction

The purpose of a digital signature is to provide a means for an entity to bind its identity to a piece of information. The process of signing transforms the message and some secret information held by the entity into a tag called a signature [5]. A verification algorithm is a method for verifying that a digital signature is authentic, i.e., was indeed created by the specified entity.

Certificateless public key cryptography introduced by Al-Riyami and Paterson [1], does not require the use of certificates to guarantee the authenticity of signer's public key. But the system parameters must be authentic. Chen et al. [3] investigated the structural extensions of security models for certificateless signatures. Bouakkaz and Semchedin [2] presented a certificateless ring signature scheme with batch verification for applications in Vehicular Adhoc Networks (VANETs).

Very recently, Feng et al. [4] have designed a certificateless signature scheme. Combining blockchain, the scheme is used for cross-domain authentication. Though the signature scheme is interesting, we find it is insecure against forgery attack. The explicit dependency between the user's certificateless public key and private key is not properly used to construct some intractable problems, such as Elliptic Curve Discrete Logarithm (ECDL), Computational Diffie-Hellman (CDH), and Decisional Diffie-Hellman (DDH). An adversary can find an efficient signing algorithm functionally equivalent to the valid signing algorithm. The findings in this note could

Z. Cao, Department of Mathematics, Shanghai University, Shanghai, 200444, China.
L. Liu, Department of Mathematics, Shanghai Maritime University, Shanghai, 201306, China.
Email: liulh@shmtu.edu.cn

be helpful for newcomers who are not familiar with the designing techniques for certificateless signature.

2 Review of Feng et al.'s signature scheme

The Feng et al.'s protocol [4] involves four entities: IIoT devices (equipped with sensors that can collect production data and communicate with edge gateways), Edge gateway, Blockchain, and KGC who is responsible for generating public parameters, and partial private keys for devices. The symbols and their descriptions are listed below (Table 1). The procedure of Feng et al.'s authentication scheme can be depicted as follows (Table 2).

Table 1: Symbols and descriptions

symbol	description
KGC	key generation center
G	A cyclic group of prime order q with a generator P
P_{pub}	system public parameter
$EGateway$	edge gateway
Dev_i	IIoT device
ID_i	Dev_i's identity
PK_i, SK_i	Dev_i's public key, secret key
P_A, S_A	$EGateway$'s public key, secret key

Table 2: The Feng et al.'s authentication scheme

System Initialization. Publish the system parameters $params = \{p, q, G, P, P_{pub}, H_1, H_2, H_3\}$.

Registration. The manager generates an identity token $Token_i^E = fac_i \| workshop_i \| equ_i \| IM_i \| P_i$ for the edge gateway in the domain and saves it on the blockchain. The process is as follows.

$$Dev_i \xrightarrow{m_1=(Register, TS_1, workshop_A, equ_A)} KGC_A \xrightarrow{m_2=(TS_2, ID_i, Q_i, d_i)}$$
$$Dev_i \xrightarrow{m_3=E_{P_A}(Upload, ID_i, PK_i, TS_3, V_1)} EGateway_A \xrightarrow{m_4=(success)} Dev_i.$$

Cross-domain Authentication. The device Dev_i in $Domain_A$ wants to establish communication with the device Dev_j in another domain $Domain_B$. They do as follows.

$$Dev_i \xrightarrow{m_5=E_{P_A}(authentication, Req_i, \sigma_i, TS_4, N_2)} EGateway_A \xrightarrow{m_6=(IM_A, Req_i, \sigma_i, TS_5, V_2, N_3)}$$
$$EGateway_B \xrightarrow{m_7=(Req_i, \sigma_i)} Verifier \xrightarrow{m_8=(success, PK_i)}$$
$$EGateway_B \xrightarrow{m_9=(TS_6, PK_i, ID_i, N_4)} Dev_j \xrightarrow{m_{10}=E_{P_B}(Req_j, TS_7, \sigma_j, V_3)}$$
$$EGateway_B \xrightarrow{m_{11}=E_{P_A}(Req_j, IM_B, TS_8, \sigma_j, V_4)} EGateway_A \xrightarrow{m_{12}=(\sigma_j, Req_j)}$$
$$Verifier \xrightarrow{m_{13}=(success, PK_j)} EGateway_A \xrightarrow{m_{14}=(ID_j, V_5, TS_9, PK_j)} Dev_i.$$

3 Analysis of the signature scheme

3.1 Inadmissible elliptic curve group

Let p be a prime number, E be an elliptic curve defined over the finite field F_p, and $E(F_p)$ be the set of F_p-rational points on E. There is a chord-and-tangent rule for adding two points in $E(F_p)$ to give a third point in $E(F_p)$. Together with this addition operation, the set of points $E(F_p)$ forms an abelian group with ∞ serving as its identity, where ∞ is the point at infinity. The number of points in $E(F_p)$, denoted $\sharp E(F_p)$, satisfies the following condition.

Theorem 1 (*Hasse interval*). *Let E be an elliptic curve defined over the finite field F_p. Then*

$$p + 1 - 2\sqrt{p} \leq \sharp E(F_p) \leq p + 1 + 2\sqrt{p}.$$

Notice that for the field F_p it is difficult to find an elliptic curve such that the associated group $E(F_p)$ has just the admissible order p. Generally, one should first choose an elliptic curve, and then choose a subgroup of $E(F_p)$. Feng et al. [4] have confused the cardinal number $\sharp(F_p)$ of finite field F_p with $\sharp(E(F_p))$ of the elliptic curve group $E(F_p)$ (see Fig.1).

> chooses a prime number q of length λ-bit, where λ is a security parameter. G is the elliptical addition group of order q in the finite field Fq,$P \in G$ is a generator, KGC randomly selects $s \in Z_q^*$ and calculates $P_{pub} = sP$, where the system master key $msk = s$ is kept secret. Select three distinct secure hash functions $H_i: \{0, 1\}^* \rightarrow Z_q^*$ $(i = 1, 2, 3)$. KGC

Figure 1: Flawed descriptions (page 521, Ref.[4])

To revise, it should introduce another prime number p and define an elliptic curve over the finite field F_p. Then choose an additive cyclic subgroup of $E(F_p)$ with the generator P of prime order q.

3.2 Some typos

The original presentation has some typos. Especially, the signature verification equation does not hold. We will correct it in the next section. Except that, the following typos should be revised.

$\diamond$ The three hash functions H_1, H_2, H_3 have the same domain and codomain. In cryptography, they have no any essential difference. So, it only needs to specify one hash function $H : \{0, 1\}^* \rightarrow Z_q^*$.

> P_A is the public key of the edge gateway, generated by running the RSA algorithm. First choose two large prime numbers k_1 and k_2, and calculate $n = k_1 * k_2$, $u = \varphi(n) = \varphi(k_1)\varphi(k_2) = (k_1 - 1)(k_2 - 1)$, after that, select an integer e that is less than u and satisfy $gcd(\phi(n), e) = 1$, and finally generate a public-private key pair $\{P_A\{e, n\}, S_A\{z, n\}\}$, where z satisfy $z = e - 1 \pmod{\phi(n)}$.

Figure 2: The screenshot for a typo (page 521, Ref.[4])

$\diamond$ In the registration phase, the description for RSA is flawed (see Fig.2). It is well-known

that in RSA cryptosystem the private key z should satisfy that

$$ez = 1 \bmod \phi(n)$$

where $\phi(\cdot)$ is the Euler's totient function, and n is the RSA modulus.

$\diamond$ The sent message (see page 523, [4]),

$$EGateway_B \to Dev_j : m_9 = E_{S_B}(TS_6, PK_i, ID_i, N_4),$$

could be revised as $m_9 = (TS_6, PK_i, ID_i, N_4)$, because the symbol E_{S_B} represents conventionally the encryption algorithm under the private key S_B. In this case, the receiver Dev_j who cannot access the private key S_B, fails to decrypt m_9.

Similarly, the sent message in the process

$$EGateway_A \to Dev_i : m_{14} = E_{S_A}(ID_j, V_5, TS_9, PK_j),$$

could be revised as $m_{14} = (ID_j, V_5, TS_9, PK_j)$, because the receiver Dev_i cannot access the private key S_A to retrieve (ID_j, V_5, TS_9, PK_j).

3.3 Insecure against forgery attack

As we know, for a certificateless signature it assumes that there are two types of attackers [3]:

- attacker $\mathcal{A}_1$ who can replace any user's certificateless public key and get any information in the public channel, but cannot access the master key;
- attacker $\mathcal{A}_2$ who can obtain the master key of TRA and any information in the public channel, but cannot change any user's certificateless public key.

The Feng et al.'s certificateless signature verification algorithm is specified as follows (Fig.3).

It is easy to find that the verification equation

$$\sigma_i T_i = h_2(T_i + P_i) + h_3(Q_i + h_1 P_{pub}) \bmod q \tag{1}$$

does not hold, because $\sigma_i = (T_i, \tau_i)$ and $\sigma_i T_i$ makes no sense. In view of that

$$\begin{aligned} &T_i = tP, \quad P_i = x_i P, \quad Q_i = rP, \quad P_{pub} = sP, \\ &d_i = r + h_1 s \bmod q, \\ &\tau_i = t^{-1}(h_2(t + x_i) + h_3 d_i) \bmod q, \end{aligned}$$

we have

$$\begin{aligned} &h_2(T_i + P_i) + h_3(Q_i + h_1 P_{pub}) \\ =&(h_2(t + x_i) + h_3(r + h_1 s))P = (h_2(t + x_i) + h_3 d_i)P \\ =&(h_2(t + x_i) + h_3 d_i)t^{-1}(tP) = \tau_i T_i. \end{aligned}$$

Algorithm 2
SC-Verify

Input: $\sigma_i = (T_i, \tau_i)$, $Req_i = fac_A||workshop_A||TS_4 \, ||ID_i, P_{pub}$
Output: success or false
1: **If** $T_{now} - TS_i < \Delta T$ **than**
2: *select* $Token_i^E$ *where* $ID = ID_i$;
3: **If** value==null;
4: return false;
5: **else**
6: return $PK_i = (Q_i, P_i)$;
7: **end if**
8: **end if**
9: $h_1 \leftarrow H_1(ID_i, Q_i, P_{pub})$;
10: $h_2 \leftarrow H_2(TID_i, T_i, PK_i)$;
11: $h_3 \leftarrow H_3(TID_i, req_i, T_i, PK_i, P_{pub})$;
12: **If**
13: $\sigma_i T_i == h_2(T_i + P_i) + h_3(Q_i + h_1 P_{pub}) \bmod q$;
14: return success;
15: **else**
16: return false;
17: **end if**

Figure 3: Signature verification algorithm (page 523, Ref.[4])

So, the original verification equation could be revised as

$$\tau_i T_i = h_2(T_i + P_i) + h_3(Q_i + h_1 P_{pub}) \tag{1'}$$

We now show that the signature scheme is insecure against certificateless public key replacement forgery attack. Given the target identity ID_i and the system public parameter P_{pub}, an adversary can pick $\theta_1, \theta_2, \theta_3 \in Z_q^*$ to replace the certificateless public key with

$$Q_i \leftarrow \theta_1 P_{pub}, \;\; P_i \leftarrow \theta_2 P_{pub}, \;\; PK_i \leftarrow (Q_i, P_i).$$

and compute the signature value $T_i = \theta_3 P_{pub}$. For any request Req_i, the adversary computes the three hash values

$$\begin{aligned} h_1 =& H_1(ID_i, Q_i, P_{pub}), \;\; h_2 = H_2(ID_i||PK_i||T_i), \\ h_3 =& H_3(Req_i||ID_i||PK_i||T_i||P_{pub}), \end{aligned}$$

and the other signature value

$$\tau_i = \theta_3^{-1}(h_2(\theta_3 + \theta_2) + h_3(\theta_1 + h_1)) \bmod q.$$

Output the forged signature (T_i, τ_i).

Clearly, the forged signature can pass the verification equation due to that

$$\begin{aligned}\tau_i T_i &= \theta_3^{-1}(h_2(\theta_3+\theta_2)+h_3(\theta_1+h_1))(\theta_3 P_{pub}) \\ &= (h_2(\theta_3+\theta_2)+h_3(\theta_1+h_1)P_{pub} \\ &= h_2(\theta_3 P_{pub}+\theta_2 P_{pub})+h_3(\theta_1 P_{pub}+h_1 P_{pub}) \\ &= h_2(T_i+P_i)+h_3(Q_i+h_1 P_{pub}).\end{aligned}$$

That means the forged signature will be accepted by the verifier.

4 Conclusion

We show that the Feng et al.'s certificateless signature scheme is insecure against forgery attack. We also corrected some typos in the original presentation. The findings in this note could be helpful for the future works on designing such certificateless signature schemes.

References

[1] S. Al-Riyami and K. G. Paterson. Certificateless public key cryptography. In Chi-Sung Laih, editor, *Proc. ASIACRYPT'03*, volume 2894 of *Lecture Notes in Computer Science*, pages 452–473, Heidelberg, 2003. Springer.

[2] S. Bouakkaz and F. Semchedine. A certificateless ring signature scheme with batch verification for applications in VANET. *J. Inf. Secur. Appl.*, 55:102669, 2020.

[3] Y. C. Chen, R. Tso, W. Susilo, X. Huang, and G. Horng. Certificateless signatures: Structural extensions of security models and new provably secure schemes. Cryptology ePrint Archive, Paper 2013/193, 2013.

[4] L. Feng, F. Qiu, K. Hu, B. Yu, J. Lin, and S. Yao. CABC: A cross-domain authentication method combining blockchain with certificateless signature for iiot. *Future Gener. Comput. Syst.*, 158:516–529, 2024.

[5] A. Menezes, P. Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, USA, 1996.