A Note on "A Time-Sensitive Token-Based Anonymous Authentication and Dynamic Group Key Agreement Scheme for Industry 5.0"

Zhengjun Cao[1], Lihua Liu[2]

Abstract. We show that the Xu et al.'s authentication and key agreement scheme [IEEE Trans. Ind. Informatics, 18(10), 7118-7127, 2022] is flawed. (1) It confused some operations for bilinear maps and presented some inconsistent computations. (2) It failed to keep anonymity, not as claimed. The adversary can use any device's public key stored in the blockchain to test some verification equations so as to reveal the identity of a target device.
Keywords: Key agreement, anonymity, authentication, blockchain.

1 Introduction

Recently, Xu et al. [1] have presented an anonymous authentication and dynamic group key agreement scheme for industry 5.0. It is designed to meet many security requirements, such as anonymity and untraceability, session key establishment, forward and backward secrecy, resistance to replay attack, impersonation attack, etc. In this note, we show that the scheme has some inconsistent computations and fails to keep anonymity, not as claimed.

2 Review of the Xu et al.'s scheme

In the proposed scenario, there are two main kinds of entities, device (DE) and private key generator (PKG). The DEs are general nodes, and have mobile capabilities. Each PKG is similar to a group controller responsible for key generation, distribution, management, and group communication tasks. Each group is dynamic, which means that DE may join or leave a group at any time. The scheme consists of seven phases: initialization, registration, authentication without token, authentication with token, group key generation, DE join, and DE leave.

Initialization. The system administrator picks a cyclic additive group G_1 with a generator Q and a cyclic multiplicative group G_2. Both are of the prime order p. Select a bilinear map $e : G_1 \times G_1 \to G_2$ and a private key s, and set the public key as $P_{pub} = sQ$. Pick two random numbers n_{1_j}, n_{2_j}, and a unique identity IDP_j for each PKG_j. Store $\{s, n_{1_j}, n_{2_j}, IDP_j\}$ in the memory of PKG_j. Publish $\{p, G_1, G_2, Q, e, P_{pub}, h(\cdot), E_k, D_k\}$. See Table I for descriptions of involved notations. For

[1]Department of Mathematics, Shanghai University, Shanghai, 200444, China
[2]Department of Mathematics, Shanghai Maritime University, Shanghai, 201306, China. liulh@shmtu.edu.cn

Table I: Notations

Symbol	Description
TID_i	Temporary identity of the DE_i
IDD_i	The identity of DE_i
IDP_j	The identity of PKG_j
GID_k	The identity of kth group
s, P_{pub}	Private key and public key of all PKGs
S_i, a_i, b_i	The DE_i's private key
W_i, A_i, B_i	The DE_i's public key
ST_i, ET_i	The authorized time slot range $[ST_i, EY_i]$
E_k, D_k	Symmetric encryption/decryption with key k
$\oplus$	Bitwise XOR operation
(a, b)	Concatenation of data a and data b
$h(\cdot)$	A hash function $h(\cdot) : \{0,1\}^* \to \{0,1\}^l$
$h^a(b)$	Perform $a+1$ hash operations on b

convenience, we now only depict the registration phase and authentication without token phase as follows (see Table II).

3 Inconsistent computations

Pairings in elliptic curve cryptography are functions which map a pair of elliptic curve points to an element of the multiplicative group of a finite field. Let n be a positive integer. Let G_1 and G_2 be Abelian groups written in additive notation. Suppose that G_1 and G_2 have exponent n (i.e., $[n]P = 0$ for all $P \in G_1, G_2$). Suppose G_3 is a cyclic group of order n written in multiplicative notation. A pairing is a function $\hat{e} : G_1 \times G_2 \to G_3$ satisfying:

> *Bilinearity.* For all $P, P' \in G_1$ and all $Q, Q' \in G_2$ we have $\hat{e}(P + P', Q) = \hat{e}(P, Q)\hat{e}(P', Q)$ and $\hat{e}(P, Q + Q') = \hat{e}(P, Q)e(P, Q')$.
>
> *Non-degeneracy.* For all $P \in G_1$, with $P \neq 0$, there is some $Q \in G_2$ such that $\hat{e}(P, Q) \neq 1$. For all $Q \in G_2$, with $Q \neq 0$, there is some $P \in G_1$ such that $\hat{e}(P, Q) \neq 1$.

To this day, the two practical examples of pairings are the Weil and Tate pairings on elliptic curves over finite fields [2]. Both use a non-rational homomorphism $\phi : G_2 \to G_1$ to construct the so-called self-pairing $e : G_1 \times G_1 \to G_3$. In view of this fact, we find the Xu et al.'s scheme have confused the basic operations for bilinear maps and presented some inconsistent computations. It wrongly specifies that

> *For points (x, y) belonging to G_1 or G_2, we only focus on x. For example, for $Q(x_Q, y_Q)$ and a private key s', we can obtain $(x_{s'Q}, y_{s'Q})$ by point multiplication operation $s'Q$, and the corresponding public key P'_{pub} is $x_{s'Q}$.*

Table II: The Xu et al.'s authentication and key agreement scheme

DE_i	Registration	PKG_j: $\{s, n_{1_j}, n_{2_j}, IDP_j\}$
Send the join request.	$\xRightarrow{\text{request}}$	Pick a unique identity IDD_i.
Store $\{IDD_i, W_i, S_i\}$.	$\xLeftarrow[\text{[secure channel]}]{IDD_i, W_i, S_i}$	Compute $W_i = h(IDD_i), S_i = sW_i$.
		Create a new block containing $\{IDD_i, W_i\}$, and link it to the Blockchain.
DE_i: $\{IDD_i, W_i, S_i\}$	Authentication	PKG_j: $\{s, n_{1_j}, n_{2_j}, IDP_j\}$
Pick random a_i, b_i and group identity GID_k. Set a timestamp T_1 and time-slot $[ST_i, ET_i]$. Compute $A_i = a_iQ, B_i = b_iQ, TK = b_iP_{pub}$, $DNT_1 \leftarrow E_{TK}(IDD_i, ST_i, ET_i, A_i)$, $DNT_2 = h(DNT_1, B_i, T_1, GID_k)S_i$.	$\xrightarrow[\text{[open channel]}]{DNT_1, DNT_2, B_i, GID_k, T_1}$	Check the timestamp T_1. Then compute $TK = sB_i$, $(IDD_i, ST_i, ET_i, A_i) \leftarrow D_{TK}(DNT_1)$. Retrieve (IDD_i, W_i) from the blockchain. Check $e(Q, DNT_2) = e(P_{pub}, h(DNT_1, B_i, T_1, GID_k)W_i)$.
	$\xleftarrow{DNT_3, DNT_4, T_2}$	If so, generate TID_i and timestamp T_2. Compute $Seeda_i = h(IDP_j, date, ST_i, ET_i, n_{1_j})$, $Seedb_i = h(IDP_j, date, ST_i, ET_i, n_{2_j})$, $S_i = sW_i, SA_i = h(IDD_i, S_i, Seeda_i, Seedb_i)$, $TS_{a_i} = h^{ST_i - 1}(Seeda_i), TS_{b_i} = h^{z - ET_i}(Seedb_i)$, $DNT_3 \leftarrow E_{TK}(TID_i, SA_i, TS_{a_i}, TS_{b_i})$,
Check the timestamp T_2. Then check $e(Q, DNT_4) = e(P_{pub}, h(DNT_3, T_2)W_i)$. If so, $(TID_i, SA_i, TS_{a_i}, TS_{b_i}) \leftarrow E_{TK}(DNT_3)$. Store $(TID_i, SA_i, TS_{a_i}, TS_{b_i}, A_i)$.		$DNT_4 = h(DNT_3, T_2)S_i$. Insert $(IDD_i, TID_i, Seeda_i, Seedb_i, SA_i, ST_i, ET_i, A_i)$ into the list L, which containing the parameters required to verify each DE's token in each PKG.

It also wrongly formulates that

$$
\begin{aligned}
&W_i = h(IDD_i), \quad S_i = sW_i, \quad DNT_4 = h(DNT_3, T_2)S_i,\\
&DNT_2 = h(DNT_1, B_i, T_1, GID_k)S_i,\\
&e(Q, DNT_2) = e(P_{pub}, h(DNT_1, B_i, T_1, GID_k)W_i),\\
&e(Q, DNT_4) = e(P_{pub}, h(DNT_3, T_2)W_i).
\end{aligned}
$$

Clearly, $W_i = h(IDD_i)$ is not a point over the underlying elliptic curve. So do DNT_2, DNT_4. Thus, the computations

$$
\begin{aligned}
&e(Q, DNT_2) = e(P_{pub}, h(DNT_1, B_i, T_1, GID_k)W_i),\\
&e(Q, DNT_4) = e(P_{pub}, h(DNT_3, T_2)W_i)
\end{aligned}
$$

make no sense. Likewise, the following computations

$$
\begin{aligned}
&e(x_Q, DNT_2) = e(x_{P_{pub}}, h(DNT_1, B_i, T_1, GID_k)W_i),\\
&e(x_Q, DNT_4) = e(x_{P_{pub}}, h(DNT_3, T_2)W_i)
\end{aligned}
$$

make no sense, too.

One should remove the above wrong specification and formulate that $W_i = h(IDD_i)Q$ i.e., converting W_i into a point over the underlying elliptic curve. In this case, all

$$DNT_2, h(DNT_1, B_i, T_1, GID_k)W_i, DNT_4, h(DNT_3, T_2)W_i$$

are compatible with the bilinear map.

4 The loss of anonymity

Anonymity is a security requirement adopted by many protocols. As for this property, it argues that (page 7124, [1]):

> *Among the messages sent during the authentication without token phase and authentication with token phase, only DNT_1, DWT_3, and HDE_i contain the IDD_i information. However, IDD_i in DWT_3 and HDE_i is protected by $h()$. In addition, if an adversary wants to get IDD_i from DNT_1, he/she must get the TK key. However, according to the computational Diffie-Hellman (CDH) problem, the adversary cannot obtain TK from P_{pub}, B_i, or Q in polynomial time.*

The argument is not sound. In fact, the legitimate PKG_j needs to decrypt DNT_1 to retrieve the identity IDD_i, and then uses it to get the target public key W_i from the blockchain. Though an adversary cannot decrypt the ciphertext, he can access the set $\Upsilon = \{(IDD_i, W_i)\}_{1\le i\le n}$, which is stored in the blockchain. The adversary who has captured $\{DNT_1, DNT_2, B_i, T_1, GID_k\}$ or $\{DNT_3, DNT_4, T_2\}$ via open channels, can test the equation

$$\begin{aligned} & e(Q, DNT_2) = e(P_{pub}, h(DNT_1, B_i, T_1, GID_k)\chi), \\ \text{or} \quad & e(Q, DNT_4) = e(P_{pub}, h(DNT_3, T_2)\chi), \ (\rho, \chi) \in \Upsilon \end{aligned}$$

Practically, the size of Υ is moderate and the success probability of such testings is not negligible. Once such a public key χ is searched out, the adversary can reveal the target identity. To achieve true anonymity, we think, one should adopt other techniques.

5 Conclusion

We show that the Xu et al.'s key agreement scheme is flawed. We hope the findings in this note could be helpful for the future work on designing such schemes.

References

[1] Z. Xu, et al.: A Time-Sensitive Token-Based Anonymous Authentication and Dynamic Group Key Agreement Scheme for Industry 5.0, *IEEE Trans. Ind. Informatics*, 18(10): 7118-7127 (2022)

[2] S. Galbraith: Pairings, *Advances in Elliptic Curve Cryptography*, London Mathematical Society Lecture Note Series 317, Cambridge University Press (2005)