A note on “a novel authentication protocol for IoT-enabled devices”

Zhengjun Cao, Lihua Liu

Abstract. We show that the authentication protocol [IEEE Internet Things J., 2023, 10(1), 867-876] is not correctly specified, because the server cannot complete its computations. To revise, the embedded device needs to compute an extra point multiplication over the underlying elliptic curve. We also find the protocol cannot provide anonymity, not as claimed. It can only provide pseudonymity.

Keywords: Authentication, anonymity, pseudonymity, point multiplication, elliptic curve.

1 Introduction

Recently, He et al. [1] have proposed a new authentication scheme for IoT-enabled devices. Its security goals include mutual authentication, device anonymity, session key agreement, perfect forward secrecy, resistance to replay attacks, DoS attacks, man-in-the-middle attacks, identity password guessing attacks, etc. In this note, we show that the protocol is not correctly specified, because the server cannot finish its computations. We also find the protocol cannot provide anonymity, instead pseudonymity. It seems that the differences between anonymity and pseudonymity are still unfamiliar to some researchers.

2 Review of the authentication protocol

In the considered scenario, there are three entities: embedded devices, server, and auxiliary server. A device authenticates and exchanges data with the server. Some necessary information is stored in the embedded device, which is assumed to be an anti-tampering device, ensuring the security of the stored data. The server stores some key information generated during the initialization phase, verifies the legitimacy of the device's identity and performs data interaction with the embedded device. The auxiliary server only stores some key information from the server, which does not directly participate in any authentication protocol but returns the key information when the server queries. The involved notations and descriptions are listed as below (see Table 1). The protocol consists of registration phase, login and authentication phase. Its procedure can be depicted as follows (see Table 2).

3 Flaws in the authentication protocol

Though the authentication protocol is interesting, we find it has two significant flaws.

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

Table 1: Notations and descriptions

ID_i	identity of device D_i	PW_i	password for device D_i
ID_s	identity of the server S	X	secret key of the server
$\oplus$	bitwise XOR operation	$\|$	concatenation operation
$T_1, \cdots, T_5$	timestamps	EXP_{time}	expiration time for a specific device
$h(\cdot)$	a hash function	G	generator of one n-order elliptic curve group

Table 2: The He et al.'s authentication protocol

$D_i : \{ID_i, PW_i\}$	$S : \{ID_s, X\}$	AS
	$\boxed{\text{Registration}}$	
Compute $I_i = h(ID_i \| PW_i)$. $\overset{I_i}{\underset{\text{[secure channel]}}{\Longrightarrow}}$	Pick a nonce R_i, compute the pseudonym $PID_i = h(R_i \| ID_s \| I_i) \oplus ID_s$, and $CK = h(R_i \| X \| EXP_{time} \| PID_i)$, $CK' = CK \times G$, $T_i = R_i \oplus h(X \| PID_i)$, $A_i = h(T_i \oplus I_i \oplus CK')$, $A'_i = A_i \times G$.	
Store $\{PID_i, CK', T_i\}$.	Store $\{A'_i, PID_i, EXP_{time}\}$. $\overset{\{PID_i, T_i, CK'\}}{\Longleftarrow}$ $\overset{\{PID_i, T_i\}}{\Longrightarrow}$	Store $\{PID_i, T_i\}$.
	$\boxed{\text{Login and Authentication}}$	
Pick a nonce $N_1 \in [2, 2^{l_h}]$. Compute $P_1 = N_1 \times G$, $P_2 = h(N_1 \times CK')$, $Y = h(P_1 \| P_2 \| T_1)$. $\overset{PID_i, P_1, Y, T_1}{\underset{\text{[open channel]}}{\longrightarrow}}$	Check the timestamp and PID_i in the database. $\overset{PID_i}{\Longrightarrow}$	Query database. $\overset{PID_i, T_i}{\Longleftarrow}$
Check the timestamp. Then compute $A_i = h(T_i \oplus I_i \oplus CK')$, $P'_4 = P_3 \times A_i$, $Z' = h(P_3 \| P'_4 \| T_3)$. If $Z' = Z$, compute $SK = h((N_1 \times P_3) \| A_i \| T_4)$, $V_i = h(SK \| (N_1 \times CK'))$. $\overset{V_i, T_4}{\longrightarrow}$	Compute $R_i = T_i \oplus h(X \| PID_i)$, $CK = h(R_i \| X \| EXP_{time} \| PID_i)$, $P'_2 = h(P_1 \times CK)$, $Y' = h(P_1 \| P'_2 \| T_1)$. If $Y' = Y$, pick a nonce $N_2 \in [2, 2^{l_h}]$. Compute $P_3 = N_2 \times G$, $P_4 = N_2 \times A'_i$, $Z = h(P_3 \| P_4 \| T_3)$. $\overset{Z, P_3, T_3}{\longleftarrow}$	
	Check the timestamp. Then compute $SK' = h((N_2 \times P_1) \| A_i \| T_4)$, $V'_i = h((P_1 \times CK) \| SK')$. Check $V'_i = V_i$.	

3.1 Inconsistent computations

As we see, the final agreed session key is set as

$$SK = h((N_1 \times P_3)\|A_i\|T_4) \tag{1}$$

for the device, and

$$SK' = h((N_2 \times P_1)\|A_i\|T_4) \tag{2}$$

for the server. Since

$$P_1 = N_1 \times G, \quad P_3 = N_2 \times G$$

we have

$$N_1 \times P_3 = N_1 \times (N_2 \times G) = N_2 \times (N_1 \times G) = N_2 \times P_1$$

Note that the timestamp T_4 is sent to the server by the device. Both two parties can access to T_4. But only the device can retrieve the term A_i by computing

$$A_i = h(T_i \oplus I_i \oplus CK')$$

The server cannot retrieve this term A_i so as to complete the computation Eq.(2), because the term $I_i = h(ID_i\|PW_i)$ is not stored in the database. Instead, the server only stores

$$\{A_i', PID_i, EXP_{time}\}$$

To fix this flaw, it should specify that

$$SK = h((N_1 \times P_3)\|A_i'\|T_4) \tag{1'}$$

$$SK' = h((N_2 \times P_1)\|A_i'\|T_4) \tag{2'}$$

In the case, the device can retrieve the term A_i' by computing

$$A_i' = A_i \times G$$

where G is the generator of underlying elliptic curve group, a public system parameter.

In the last stage, the device needs to compute the verifier

$$V_i = h(SK\|(N_1 \times CK')) \tag{3}$$

while the server computes

$$V_i' = h((P_1 \times CK)\|SK') \tag{4}$$

It is easy to find that

$$\begin{aligned} V_i &= h(SK\|(N_1 \times CK')) \\ &\neq h((P_1 \times CK)\|SK') = V_i' \end{aligned}$$

due to the collision-free property of the hash function. To fix the flaw, it can specify the server's verifier as

$$V_i' = h(SK'\|(CK \times P_1)) \tag{4'}$$

owing to

$$N_1 \times CK' = N_1 \times (CK \times G) = CK \times (N_1 \times G) = CK \times P_1$$

3.2 The loss of anonymity

In cryptography, anonymity refers to the state of being completely nameless, with no attached identifiers. Pseudonymity involves the use of a fictitious name that can be consistently linked to a particular user, though not necessarily to the real identity [2]. Both provide a layer of privacy, shielding the user's true identity from public view. However, the key difference lies in traceability. While anonymous actions are designed to be unlinkable to any one individual, pseudonymous actions can be traced back to a certain entity.

We want to stress that the true anonymity means that the adversary cannot attribute different sessions to target entities. In other words, it relates to entity-distinguishable feature, not just identity-revealable feature.

In the He et al.'s authentication protocol, the embedded device with the identity ID_i needs to send the data

$$\{PID_i,\ P_1,\ Y,\ T_1,\ V_i,\ T_4\}$$

to the server via an open channel. An adversary can capture the pseudonym PID_i and recognize the target device by checking the consistency of this pseudonym. In nature, this protocol can only provide pseudonymity, not the usual anonymity.

4 Conclusion

We show that the He et al.'s authentication scheme cannot provide anonymity, and clarify the differences between anonymity and pseudonymity. We also correct some inconsistent computations in the original presentation. We hope the findings in this note could be helpful for the future work on designing such schemes.

References

[1] D. He, Z. Zhao, S. Chan, and M. Guizani: A Novel Authentication Protocol for IoT-Enabled Devices. *IEEE Internet Things J.*, 2023, 10(1), 867-876.

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