On the Anonymity in "A Practical Lightweight Anonymous Authentication and Key Establishment Scheme for Resource-Asymmetric Smart Environments"

Zhengjun Cao, Lihua Liu

Abstract. We show that the anonymous authentication and key establishment scheme [IEEE TDSC, 20(4), 3535-3545, 2023] fails to keep user anonymity, not as claimed. We also suggest a method to fix it.

Keywords: Mutual authentication, Key agreement, Rabin cryptosystem, User anonymity.

1 Introduction

Recently, Bai et al. [1] presented an anonymous authentication and key establishment scheme for resource-asymmetric smart environments. It is designed to meet many security requirements, including mutual authentication, secure session key agreement, user anonymity and untraceability, forward and backward security, resistance to mobile device loss attacks, impersonation attacks, privileged-insider attack, replay attacks, man-in-the-middle attacks, and offline guessing attacks. In this note, we show that the scheme fails to keep user anonymity, not as claimed. We also suggest a remedy.

2 Review of the authentication and key agreement protocol

In the considered scenario, there are three entities: user, gateway, and smart device (SD). SD is in a smart environment, containing sensitive data. Only authenticated users can obtain data from SD. The user can communicate with gateway node to obtain SD data. Before accessing SD data, user should register with the third-party. User's real ID is encrypted by Rabin cryptosystem [2].

The protocol consists of four phases: system initialization phase, registration phase, login and authentication phase, and modify password offline phase. The third-party initializes the entire system, by selecting two large primes P, Q to set $N = P * Q$, and a hash function $H(\cdot) : \{0,1\}^* \to \{0,1\}^\ell$ where the positive integer ℓ is a security parameter. Publish N and $H(\cdot)$. The involved notations are listed below (Table 1).

The registration and session key agreement phases can be depicted as follows (Table 2).

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

Table 1: Notations and descriptions

Notation	Description
N	RSA modulus
$H(\cdot)$	hash function
$\|$	string concatenation operator
$\oplus$	bitwise XOR operator
SD	smart device/sensor
SID_k	SD identification
RabinDec($\cdot$)	Rabin decryption function
ID_i, PW_i	user identification and password
$A_i, B_i, C_i, R_a, R_b, R_c$	random numbers

3 Loss of user anonymity

In the considered scenario, it assumes that the adversary A may be a legitimate user, knowing the entire steps of the key establishment scheme, and having the capabilities [1]:

- Be able to intercept, forge, delete and replay the information transmitted in the public channel, to get the parameters transmitted during the registration phase from the third-party, to steal the parameters stored in the user device, to get the session key used in the past, but only get one of user device and user password.
- *Master identity space and password space, and can enumerate all the correspondences.*

In order to mask the user's identity ID_i, it adopts the following transformations:

$$\begin{array}{ll} \text{(pseudonym)} & DID_i = (ID_i \| t_1)^2 \bmod N, \\ \text{(real identity)} & ID_i = \text{RabinDec}(P, Q, DID_i) \end{array}$$

where the pseudonym DID_i and the timestamp t_1 are sent via the open channel. So, the adversary can obtain DID_i, t_1 by eavesdropping the channel. Besides, by the threat model assumption, we know, the adversary can access to the identity space Υ and the system public parameter N. Therefore, the adversary can choose any identity λ to test

$$DID_i \overset{?}{=} (\lambda \| t_1)^2 \bmod N, \quad \lambda \in \Upsilon \tag{1}$$

Once the equation holds, the target identity ID_i will be retrieved. If the size of identity space Υ is moderate, the success probability of above testing is not negligible.

4 A remedy

In order to estimate the protocol's communication overhead, it assumes that (see §6.2, Ref.[1])

- the length of timestamp is 32 bits;
- the length of the user identity, user password and sensor identity are 128 bits;

Table 2: The Bai et al.'s authentication and key agreement protocol

User (U_i)	Third-party	SD
Choose ID_i, PW_i. Generate random A_i. Compute $HPW_i = H(A_i \Vert PW_i)$. $\xRightarrow[\text{[secure channel]}]{ID_i,\ HPW_i}$	Generate random B_i. Compute $E_i = HPW_i \oplus B_i$. Store $\{ID_i, B_i, HoneyList\}$ into the gateway's memory. $\xLeftarrow{E_i}$	Send $SIDK$ $\xLeftarrow{SIDK}$
Compute $T_i = A_i \oplus H(ID_i \Vert PW_i)$, $Auth_i = H(A_i \Vert B_i) \bmod M$. Store $(E_i, T_i, Auth_i)$.	Generate random C_i. Store $(SIDK, C_i)$ into the gateway's memory. $\xRightarrow{C_i}$	Store C_i.
User	Gateway	SD
Input ID_i, PW_i. Compute $A_i = H(ID_i \Vert PW_i) \oplus T_i$, $HPW_i = H(A_i \Vert PW_i)$, $B_i = HPW_i \oplus E_i$, $Auth'_i = H(A_i \Vert B_i) \bmod M$. If $Auth'_i = Auth_i$, generate a timestamp t_1 and random R_a. Compute $DID_i = (ID_i \Vert t_1)^2 \bmod N$, $M_0 = (SIDK \Vert R_a) \oplus B_i$, $Verify1 = h(ID_i \Vert B_i \Vert R_a \Vert t_1)$. $\xrightarrow[\text{[open channel]}]{DID_i,\ M_0,\ t_1,\ Verify1}$	Check the timestamp t_1. Compute $ID'_i = \text{RabinDec}(P, Q, DID_i)$, $B_i = \text{get}(ID_i, \text{UserTable})$, $(SIDK', R'_a) = M_0 \oplus B_i$, $Verify1' = H(ID'_i \Vert B_i \Vert R'_a \Vert t_1)$. If $Verify1' = Verify1$, generate random R_b, timestamp t_3. Compute $C_i = \text{get}(SIDK', \text{SensorTable})$, $SKt1 = H(ID'_i \Vert R'_a \Vert R_b)$, $M_1 = (R_b \Vert SKt1) \oplus C_i$, $Verify2 = H(SKt1 \Vert SIDK' \Vert t_3)$. $\xrightarrow{M_1,\ t_3,\ Verify2}$	Check the timestamp t_3. Compute $(R'_b \Vert SKt1') = M_1 \oplus C_i$, $Verify2' = H(Skt1' \Vert SIDK \Vert t_3)$. If $Verify2' = Verify2$, generate random R_c, timestamp $t5$. Compute $SKt2 = H(SIDK \Vert R_c \Vert R'_b)$, $SK = SKt1' \oplus SKt2$, $M_2 = R_c \oplus C_i$, $Verify3 = H(R_c \Vert SK \Vert t_5)$. $\xleftarrow{M_2,\ t_5,\ Verify3}$
Check the timestamp t_7. Compute $(R'_b \Vert SKt2) = M_3 \oplus B_i$, $Verify4' = H(SKt2 \Vert B_i \Vert t7)$. If $Verify4' = Verify4$, compute $SKt1 = H(ID_i \Vert R_a \Vert R'_b)$, $SK = SKt1 \oplus SKt2$.	Check the timestamp t_5. Compute $R'_c = M_2 \oplus C_i$, $SKt2 = H(SIDK \Vert R'_c \Vert R_b)$, $SK = SKt1 \oplus SKt2$, $Verify3' = H(R'_c \Vert SK \Vert t_5)$. Check if $Verify3' = Verify3$. Generate timestamp $t7$. Compute $M_3 = (R_b \Vert SKt2) \oplus B_i$, $Verify4 = H(SKt2 \Vert SK \Vert B_i \Vert t_7)$. $\xleftarrow{M_3,\ t_7,\ Verify4}$	

- the length of secret key, random number, hash function, and message authentication code are 160 bits;
- the length of modular exponentiation operation is 1024 bits.

Namely, we have

$$DID_i = (\underbrace{ID_i}_{\text{128-bit}} \| \underbrace{t_1}_{\text{32-bit}})^2 \bmod \underbrace{N}_{\text{1024-bit}} \tag{2}$$

The square $(ID_i\|t_1)^2$ is strictly less than the modular N, i.e.,

$$DID_i = (ID_i\|t_1)^2 \tag{3}$$

which is a common equation. The adversary can recover $ID_i\|t_1$ by general calculations.

Based on this observation, we now suggest a method to fix the loss of user anonymity. To generate the pseudonym, the user can compute

$$\begin{aligned} DID_i =(&\underbrace{ID_i}_{\text{128-bit}} \| \underbrace{t_1}_{\text{32-bit}} \| \underbrace{A_i}_{\text{160-bit}} \| \underbrace{B_i}_{\text{160-bit}} \| \\ &\underbrace{H(A_i\|B_i\|t_1)}_{\text{160-bit}} \| \underbrace{H(A_i\|PW_i\|t_1)}_{\text{160-bit}})^2 \bmod N \end{aligned}$$

In this case, the plaintext is of 800 bits. It is a true congruence equation, not a common equation. An adversary cannot construct a testing equation for any target identity even if the identity space Υ has a moderate size, because the substring

$$A_i\|B_i\|H(A_i\|B_i\|t_1)\|H(A_i\|PW_i\|t_1)$$

is strictly unaccessible to the adversary.

5 Conclusion

We show that the Bai et al.'s authentication and key agreement scheme cannot provide user anonymity. We also suggest a method to fix the flaw. The analysis techniques developed in the note could be helpful for the future work on designing such schemes.

References

[1] L. Bai, C. Hsu, L. Harn, J. Cui, Z. Zhao: A practical lightweight anonymous authentication and key establishment scheme for resource-asymmetric smart environments, *IEEE Trans. Dependable Secur. Comput.*, 20(4), 3535-3545 (2023)

[2] M. Rabin: Digitalized signature as intractable as factorization, *technical report MIT/LCS/TR-212, MIT Laboratory for Computer Science*, January (1978)