A note on "a lightweight mutual authentication and key agreement protocol for remote surgery application in Tactile Internet environment"

Zhengjun Cao[1], Lihua Liu[2]

Abstract. We show that the key agreement scheme [Comput. Commun., 2021(170): 1–18] is insecure against impersonation attacks, because there is a trivial equality which results in the loss of data confidentiality.

Keywords: Mutual authentication, Key agreement, Impersonation attack, Data confidentiality

1 Introduction

Recently, Kamil and Ogundoyin [1] have presented a mutual authentication and key agreement scheme for Tactile Internet-assisted remote surgery application, in which there are four entities: a fully trusted entity (TA), devices (Robots) installed in the theater operation room of the hospital, a node (Gateway) which acts as an intermediary between a remote surgeon and a robot, and a registered and authorized medical practitioner (Surgeon or User) who conducts surgical operation from a remote location using some robotic arms placed in the theater room where the patient is positioned. The TA is responsible for initialization. Surgeons and robots register with TA via secure communication channels, respectively. Then Surgeon and Robot will mutually authenticate with each other by using key agreement through public channel. The scheme only involves lightweight operations, such as hashing, string concatenation, and bit-wise XOR. Though the scheme is interesting, we find it flawed.

2 Review of the scheme

Let $h : \{0,1\} \to Z_q^*$ be a hash function, where q is a big prime. The bit-wise XOR operation is denoted by $\oplus$, and the concatenation operation is denoted by $\|$. $\triangle T$ represents the allowable network transmission delay.

The scheme consists of seven phases: Gateway and Robotic Arm Registration, User Registration, User Login, Authentication and Key Agreement, Password Updating, Dynamic Robotic Arm Addition, and Revocation. The related phases can be described as follows (see Table 1).

[1] Department of Mathematics, Shanghai University, Shanghai, 200444, China

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

Table 1: The Kamil-Ogundoyin key agreement scheme

Gateway G_i	$\boxed{\text{Gateway registration}}$	TA
	$\underset{\text{[secure channel]}}{\overset{\{RID_i,D_i,RID_j,D_j\}}{\Longleftarrow}}$	Pick a unique identity RID_{TA}, RID_i as the identity of G_i. a secret $s \in Z_q^*$, to compute $D_i = h(s, RID_{TA}, RID_i)$. Store $\{RID_i, D_i, RID_j, D_j\}$ in G_i's memory.
Robot RM_j	$\boxed{\text{Robot registration}}$	TA
	$\overset{\{RID_j,D_j\}}{\Longleftarrow}$	Pick RID_j as RM_j's identity. Compute $D_j = h(s, RID_{TA}, RID_j)$. Store $\{RID_j, D_j\}$ in RM_J's memory.
Surgeon S_k	$\boxed{\text{Surgeon registration}}$	TA
Pick identity RID_k, nonce B_k, password PW_k. Compute $D_k = h(RID_k, B_k)$, $HPW_k = h(PW_k, B_k)$.	$\overset{D_k,HPW_K}{\Longrightarrow}$	Pick a nonce C, compute $\alpha = h(C, D_i) \oplus h(D_k, HPW_k)$, $\beta = C \oplus h(RID_i, D_i)$. Store $\{\alpha, \beta\}$ in a mobile device.
Compute $A_2 = h(B_k, HPW_k, D_k)$, $A_1 = h(PW_k, RID_k) \oplus B_k$. Store $\{A_1, A_2\}$ into the device.	$\overset{\{\alpha,\beta\}}{\Longleftarrow}$	
Surgeon S_k: $\{\alpha, \beta, A_1, A_2\}$	Gateway G_i: $\{RID_i, D_i, RID_j, D_j\}$	Robot RM_j: $\{RID_j, D_j\}$
	$\boxed{\text{Login \& authentication \& key agreement}}$	
Input identity RID_k, password PW_k. Compute $B_k = A_1 \oplus h(PW_k, RID_k)$, $D_k = h(RID_k, B_k)$, $HPW_k = h(PW_k, B_k)$. Check $A_2 = h(B_k, HPW_k, D_k)$. If so, the device picks nonce R_k, time stamp TS_1, to compute $A_3 = \alpha \oplus h(D_k, HPW_k)$, $A_4 = \beta \oplus TS_1$, $A_5 = h(R_k, A_3, TS_1)$, $A_6 = (R_k \| A_5) \oplus A_3$.	Check $TR_1 - TS_1 < \triangle T$. Compute $C = A_4 \oplus h(RID_i, D_i) \oplus TS_1$, $A_3 = h(C, D_i)$, $R_k \| A_5 = A_6 \oplus A_3$.	
$\underset{\text{[public channel]}}{\xrightarrow{A_4,A_5,A_6,TS_1}}$	Check $A_5 = h(R_k, A_3, TS_1)$	
	Pick nonce R_i, time stamp TS_2, compute $A_8 = h(RID_j, D_j, C, R_i, TS_2)$, $A_7 = C \oplus h(RID_j, D_j, R_j, R_k, TS_2)$, $A_9 = D_j \oplus (R_i \| R_k \| TS_2)$.	Compute $R_i \| R_k \| TS_2 = A_9 \oplus D_j$. Check $TR_2 - TS_2 < \triangle T$. Compute
	$\xrightarrow{A_7,A_8,A_9}$	$C = A_7 \oplus h(RID_j, D_j, R_i, R_k, TS_2)$. Check $A_8 = h(RID_j, D_j, C, R_i, TS_2)$. Pick nonce R_j, time stamp TS_3, compute $K_1 = h(R_i, R_k, R_j)$, $A_{10} = h(R_i, R_j, K_1, RID_j, D_j, TS_3)$, $A_{11} = R_i \oplus (R_j \| TS_3)$.
	Compute $R_j \| TS_3 = A_{11} \oplus R_i$. Check $TR_3 - TS_3 < \triangle T$.	$\xleftarrow{A_{10},A_{11}}$
Compute $R_i \| R_j \| TS_4 = A_{13} \oplus R_k$.	Compute $K_2 = h(R_i, R_k, R_j)$. Check $A_{10} = h(R_I, R_j, K_2, RID_j, D_j, TS_3)$. Compute $A_{12} = h(K_2, R_i, R_j, A_8, TS_4)$, $A_{13} = (R_i \| R_j \| TS_4) \oplus R_K$. $\xleftarrow{A_8,A_{12},A_{13}}$	
Check $TR_4 - TS_4 < \triangle T$. Compute $K_3 = h(R_i, R_k, R_j)$. Check $A_{12} = h(K_3, R_i, R_j, A_8, TS_4)$.		

3 Impersonating a robot

In the discussed model, the communications between any two entities are done over an open channel and the endpoint parties such as robots and surgeons are considered as untrustworthy. It claims that (see §3.2, [1]): *The adversary $\mathcal{A}$ can physically capture a robotic arm and extract all the secret credentials stored in its memory. $\mathcal{A}$ can also clone the captured robot and replace it with a malicious one.*

However, if the robot RM_j is compromised and RID_j, D_j are extracted, then $\mathcal{A}$ can directly impersonate RM_j to cheat the surgeon S_k. In fact, the following computations

$$\begin{aligned}
&R_i \| R_k \| TS_2 = A_9 \oplus D_j \\
&C = A_7 \oplus h(RID_j, D_j, R_i, R_k, TS_2) \\
&A_8 = h(RID_j, D_j, C, R_i, TS_2) \\
&A_{10} = h(R_i, R_j, K_1, RID_j, D_j, TS_3)
\end{aligned}$$

invoke RID_j and D_j in a very simple way, not relying on any exquisite logical relationship. Besides, in the registration phase the TA's master secret key s is superficially invoked to compute

$$D_j = h(s, RID_{TA}, RID_j)$$

instead of being bound to other parameters. The designing bug results in the impersonation attack. To thwart this attack, the identity RID_j cannot be stored in RM_j's memory so as to prevent an adversary from extracting it.

4 Impersonating a surgeon

As we know, the Boolean logic operation XOR is widely used in cryptography which compares two input bits and generates one output bit. When the operator is performed on two strings, they must be of a same bit-length. Otherwise, the shorter string should be stretched by padding some 0s to its left side. In this case, the partial string corresponding to the padding bits is eventually exposed to the adversary.

In the discussed scheme,

$$\begin{aligned}
&\alpha = h(C, D_i) \oplus h(D_k, HPW_k) \\
&A_3 = \alpha \oplus h(D_k, HPW_k) \\
&A_5 = h(R_k, A_3, TS_1)
\end{aligned}$$

A_3 and A_5 have a same bit-length, which equals to that of any output of hash function h. Therefore, $A_6 = (R_K \| A_5) \oplus A_3$ is a trivial equality. Naturally,

$$A_6 = R_K \| (A_5 \oplus A_3) \tag{1}$$

results in the loss of data confidentiality, in which the nonce R_K is directly exposed.

By the captured $\{A_4, A_5, A_6, TS_1\}$, an adversary $\mathcal{A}$ can first parse A_6 to recover R_K and $A_5 \oplus A_3$.

Then retrieve

$$A_3 = (A_5 \oplus A_3) \oplus A_5 \tag{2}$$

Notice that $A_3 = H(C, D_i)$ is unchanged in different sessions launched by the surgeon.

With the retrieved A_3, the adversary can impersonate the surgeon to cheat the gateway G_i and the robot RM_j. To do so, the adversary $\mathcal{A}$ picks a nonce R'_k to compute

$$A'_4 = A_4 \oplus TS_1 \oplus TS'_1 = \beta \oplus TS'_1$$
$$A'_5 = h(R'_k, A_3, TS'_1), \; A'_6 = (R'_k \| A'_5) \oplus A_3$$

where TS'_1 is a new time stamp. Then send the message $\{A'_4, A'_5, A'_6, TS'_1\}$ to the gateway G_i. Clearly, the new message can pass the unique verification

$$A'_5 = h(R'_k, A_3, TS'_1) \tag{3}$$

performed by G_i in the first stage.

By the way, the transformations

$$A_9 = D_j \oplus (R_i \| R_k \| TS_2) \tag{4}$$

$$A_{13} = (R_i \| R_j \| TS_4) \oplus R_K \tag{5}$$

are insufficient to mask the substrings $R_i \| R_k$ and $R_i \| R_j$, respectively, depending on the effective bit-length of time stamps TS_2 and TS_4.

5 Conclusion

In this note, we show that the Kamil-Ogundoyin key agreement scheme is flawed because it is not explicitly organized. The findings in this note could be helpful for the future work on designing such key agreement schemes.

References

[1] I. Kamil, S. Ogundoyin: A lightweight mutual authentication and key agreement protocol for remote surgery application in Tactile Internet environment. Comput. Commun., 170 (2021): 1–18.