Hindawi
Security and Communication Networks
Volume 2021, Article ID 9029664, 14 pages
https://doi.org/10.1155/2021/9029664

Research Article

Security and Privacy for Edge-Assisted Internet of Things Security Proof for the SKKE Protocol

Xiangyang Wang,[1] **Chunxiang Gu**,[1,2] **Fushan Wei**,[1] **and Siqi Lu**[1]

[1]*Henan Key Laboratory of Network Cryptography Technology, Zhengzhou, China*
[2]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China*

Correspondence should be addressed to Siqi Lu; 080lusiqi@sina.com

Received 5 July 2021; Accepted 9 November 2021; Published 3 December 2021

Academic Editor: Ding Wang

As an Internet of things (IoT) technology, the ZigBee has a wide range of applications in home automation, smart energy, commercial building automation, personal, home and hospital care, telecom, and wireless sensor. The ZigBee standard has the advantage of high reliability, which is based on the security of authentication key agreement protocol, namely, the SKKE protocol. In the ZigBee standard, this protocol based on shared symmetric-key is applied on the security protocol level. It is a full symmetric-key key agreement with key confirmation scheme, while the key confirmation mechanism is provided by a message authentication coding mechanism. In this paper, we consider the security of the SKKE protocol. In the random Oracle model, we reduce the security of the SKKE protocol to the collision of the hash function and the HMAC function and the indistinguishability between the output of the random Oracle and a random number. We also give a theoretical proof with the game-based method. To our knowledge, there is no research on the provable security of the ZigBee protocol at this stage, so it is helpful to promote further research of the ZigBee protocol security.

1. Introduction

The emerging IoT technology usage in different aspects of life results in a very large number of devices connected to each other and to the Internet in a small space. As we have known, IoT devices rely on one or more communication technologies [1–3] (such as ZigBee, Z-Wave, BLE, and WiFi), some of which are well-suited for resource-constrained devices to operate in low-power and lossy networks (LLNs) such as ZigBee and Z-Wave. Among them, the ZigBee protocol, with its low-cost, low-power consumption, high-robustness, and flexibility, has been increasingly applied in the field of short distance communication, daily monitoring, and short distance data transmission. Nowadays, ZigBee-based IoT devices are predominant in the IoT landscape, and the security of the ZigBee plays a more and more important role in the ZigBee communication. In recent years, the researches of the ZigBee security can be divided into implementation level security and specification level security.

At the implementation level, some researchers have studied ZigBee enabled devices in order to find vulnerabilities at the implementation level [4, 5]. These attacks found in the research include replay attack, eavesdropping attack, acquiring key attack, denial of service attack, same-nonce attack, ghost attack, and men-in-the-middle attack. However, these findings do not necessarily reflect the insecurity in the specification. In addition, their methods are usually special and aim to discover low-level vulnerabilities and cannot be effectively transported to the formal analysis of the protocol stack, while, at the specification level, formal verification of ZigBee has been considered in several papers [6–11]. Especially, in [12], the authors focused on the application of formal verification to protocols in the IoT domain and summarized the existing approaches to the formal analysis of the ZigBee protocol. In addition, Li et al. in [13] established the ZigBee symbol model according to the ZigBee standard (ZigBee 1.0 and ZigBee 3.0). This model mainly captures the important elements, such as key sharing,

device joining, and key updating. Next, according to the ZigBee specification, the authors divided the ZigBee protocol into some subprotocols, then the authors used the security protocol verification tool (tamarin) to analyze and verify the security attributes of those subprotocols. Security analysis finds some known security vulnerabilities in ZigBee 1.0 and proves that there are no such vulnerabilities in ZigBee 3.0.

Since the 1980s, two methods for analyzing security protocols have been developed [14]. One method relies on the symbolic model of protocol execution, that is, symbolic (or Dolev Yao or formal) method, in which cryptographic primitives are regarded as black boxes. Symbolic method adopts a highly abstract execution view, in which the messages exchanged by all parties are symbolic terms. This model supports automatic analysis, but the high degree of abstraction makes the security guarantee provided by this method unclear and may miss the possible attacks in the computing model. The other method relies on the computational model which considers the complexity and probability problems, namely, computational (or cryptographic) method. This method captures a powerful security concept and can resist all probabilistic polynomial-time attacks.

All above researches of the ZigBee security at the specification level are in symbolic methods, so it is very necessary and meaningful to take research on the ZigBee security at the specification level under the computational model. In the ZigBee specification, the SKKE protocol is used for two end devices (not including trust center) to establish the application link key for their communication end-to-end (only used in APS layer). Although the process of establishing the end-to-end application link key has been researched in [13], the authors only considered the case that the trust center randomly generates the link key and then distributes it to the initiator and the responder, respectively. As for another case, the initiator and the responder share the master key (randomly generated by the trust center and transmitted to the initiator and the responder, respectively), and then they negotiate the application link key based on the shared master key through the SKKE protocol [15, 16]. The SKKE protocol is a full symmetric-key key agreement with key confirmation mechanism, while message authentication coding mechanism is used to provide key confirmation. Yuksel et al. [15, 16] described the SKKE protocol in detail, but the analysis of the SKKE protocol security is scarce at present. It is necessary to analyze and prove the security of the SKKE protocol. In this paper, the security model of the SKKE protocol is established under the random Oracle model, and its confidentiality is reduced to the collision of hash function and HMAC function and the indistinguishability between random Oracle output and random number. Meanwhile, its authentication is reduced to the collision between hash function and HMAC function, and the proof of the SKKE protocol confidentiality and authentication is given with the game-based method.

The rest of this paper is organized as follows. Section 2 gives the basic knowledge needed in the security proof of the SKKE protocol; Section 3 is the description of the SKKE protocol and its execution details; in Section 4, we model the SKKE protocol and give the definition of indistinguishable experiment, authentication experiment, and some security notions; in Section 5, we prove the security of the SKKE protocol with the game-based method; Section 6 is the summary of our work.

2. Preliminaries

Denote by $x \longleftarrow y$ the operation of assigning y to x. Denote by $x \xleftarrow{\$}$ the operation sampling x uniformly at random from a set X. *PPT* is short for probabilistic polynomial time.

Lemma 1 (see [17]). *Let $\mathscr{A}$ be a distinguisher, then we have*

$$\begin{aligned} &|\Pr[\mathscr{A}(y) = 1 | f \in F^{x,y}, y = f(x)] \\ &\quad - \Pr[\mathscr{A}(y) = 1 | y \xleftarrow{\$} R]| \leq \mathrm{negl}(n), \end{aligned} \tag{1}$$

where $F^{x,y}$: $x \longmapsto y$ is a function cluster; $|x| = |y| = n$; $f \in F^{x,y}$ is a random Oracle; $y \xleftarrow{\$}$ denotes sampling y uniformly at random from R; and $\mathrm{negl}(n)$ is a negligible function.

HMAC [18] is a key-related hash operation message authentication code. HMAC operation uses a hash algorithm to generate a message digest as output with a key and a message as input. A key hash function and a key are used in the definition of HMAC. "text" is used as the plaintext to calculate HMAC; the operation of HMAC function is as follows:

$$H'\left((K \oplus \mathrm{opad}) \| H'(K \oplus \mathrm{ipad} \| \mathrm{text})\right), \tag{2}$$

where H' is a hash function; ipad and opad are two different fixed strings. The detailed operation steps of HMAC function can be referred to [18].

3. The Description of the SKKE Protocol

In this section, we describe the interaction and the execution details of the SKKE protocol.

3.1. The SKKE Protocol. Figure 1 shows the message transmission of the key agreement mechanism, namely, the SKKE protocol [16]. As shown in the figure, U represents the initiator of the protocol and V represents the responder of the protocol. The essential difference between the role of the initiator and the role of the responder is that the initiator sends the first pass of the exchange. As shown in Figure 1, the MK is the master key shared between the initiator U and the responder V. $A\langle B\rangle$ means the transmission of message A that contains payload(s) B. QEX represents the challenge generated by a device X and MacTag represents the hash value. Based on the master key MK, U and V negotiate the session key LK by executing the SKKE protocol.

3.2. Execution Details of the SKKE Protocol. In [15, 16], the authors showed a detailed description of the SKKE protocol. For convenience, let us briefly describe it here.

In the SKKE protocol, an initiator U establishes a LK with a responder V using a shared secret MK. We present the computational details of the SKKE protocol in Figure 2.

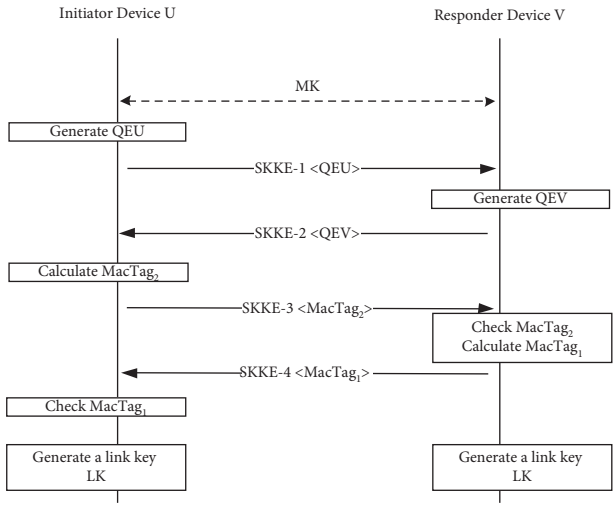

Figure 1: Symmetric-key authenticated key agreement scheme.

In the first two messages, the initiator and the responder exchange their challenges. In the last two messages, they exchange the data they have computed using the challenges and their identities. After verifying that they have received the correct value, they use another value as the LK that both of them can compute.

4. Security Model and Security Experiment

4.1. Security Model. We assume that the attacker can completely control the channel, that is, the attacker can eavesdrop, delay, and tamper with the messages transmitted by running the protocol. The following Oracle queries are defined to describe the adversary's ability:

(i) Execute (U, V): this query models passive attacks. The output of this query consists of messages that were exchanged during the honest execution of the protocol among U and V.

(ii) Send (U, m): this query models active attack by sending message m to the participant U. The output of this query is the message that U would generate on receipt of message m.

(iii) Reveal (U_i, V_i): this query models the misuse of session key, namely, link key established between U and V. The output of this query is the link key LK_i established during the i-th communication between U and V. If the i-th communication is used as the test session, this query cannot be executed.

(iv) Corrupt (U_i, V_i): this query models the ability of the adversary A to reveal the long-term secret key of the i-th communication between U and V. The output of this query is the master key MK_i shared between U and V during the i-th communication. If the i-th communication is used as the test session, this query cannot be executed.

The adversary A is represented as two procedures A_1 and A_2 that share state. In the RO model, the hash functions H and H' are modeled as two random oracles, H_A and H'_A, and each of them obtains a list to store queries. When the adversary A queries to H_A, if the query is in the list, the corresponding value is returned, else it returns a random value. It is the same as above that the adversary A queries to H'_A.

4.2. Security Experiment. Before defining the security notion of the SKKE protocol, we define indistinguishability experiment and authentication experiment from the perspectives of confidentiality and authentication of the SKKE protocol.

4.2.1. Indistinguishability Experiment. In this section, we define an indistinguishability experiment (see Figure 3), which is used in the security proof of confidentiality.

The (U_i, V_i) represents i-th $(i = 1, 2, \ldots, N)$ session established by the initiator U_i and the responder V_i, which is the test session and not revealed. MK_i and LK_i represent the master key and the link key, respectively, which is a bijection. keylen represents the length of the master key and the link key.

Definition 1. For any *PPT* adversary $\mathcal{A}$, the advantage of the adversary $\mathcal{A}$ breaks the confidentiality of the SKKE protocol which is defined as follows:

$$Adv^{ind}_{\mathcal{A}}(l, N, q_H, q_{H'}) = \left|\Pr\left[Exp^{ind}_{\mathcal{A}}(l, N, q_H, q_{H'}) = 1\right] - \frac{1}{2}\right|. \tag{3}$$

4.2.2. Authentication Experiment. We denote the authentication experiment as $\mathrm{Exp}^{\mathrm{auth}}_{\mathcal{A}}(l, N, q_H, q_{H'})$. Next, we consider the authentication of the initiator U_i to the responder V_i, namely, $\mathrm{Exp}^{\mathrm{auth\text{-}uv}}_{A}(l, N, q_H, q_{H'})$ (see Figure 4), as an example to consider the authentication of the SKKE protocol.

Remark 1. The SKKE protocol is a symmetrical two-way authentication protocol. In Figure 4, $\mathrm{Exp}^{\mathrm{auth\text{-}uv}}_{A}(l, N, q_H, q_{H'})$ denotes the authentication of the initiator U_i to the responder V_i, as for the authentication of the responder V_i to the initiator U_i, similarly.

Definition 2. For any *PPT* adversary A, the advantage of the adversary A breaks the authentication of the SKKE protocol which is defined as follows:

$$Adv^{auth}_{\mathcal{A}}(l, N, q_H, q_{H'}) = \Pr\left[Exp^{auth}_{\mathcal{A}}(l, N, q_H, q_{H'}) = 1\right]. \tag{4}$$

Definition 3. The SKKE protocol has security if for any *PPT* adversary A, the following advantage is negligible:

		U	V
SKKE-1	$U \to V$	QEU	
SKKE-2	$V \to U$	QEV	
Computation		$Z = MAC\{U \| V \| QEU \| QEV\}_{MK}$ $MacKey = H(Z \| 0x00000001)$ $KeyData = H(Z \| 0x00000002)$ $MacData_2 = 0x03 \| U \| V \| QEU \| QEV$ $MacTag_2 = MAC(MacData_2)_{MacKey}$	$Z = MAC\{U \| V \| QEU \| QEV\}_{MK}$ $MacKey = H(Z \| 0x00000001)$ $KeyData = H(Z \| 0x00000002)$ $MacData_1 = 0x02 \| V \| U \| QEV \| QEU$ $MacTag_1 = MAC(MacData1)_{MacKey}$
SKKE-3	$U \to V$	$MacTag_2$	
SKKE-4	$V \to U$	$MacTag_1$	
Verification		$MacData_1 = 0x02 \| V \| U \| QEV \| QEU$ Verify $MacTag_1$ using $MacData_1$	$MacData_2 = 0x03 \| V \| U \| QEV \| QEU$ Verify $MacTag_2$ using $MacData_2$
U and V use $KeyData = H(MAC\{U \| V \| QEU \| QEV\}_{MK} \| 0x00000002)$ as the new LK			

Figure 2: The SKKE protocol (H, hash function MAC, HMAC function; ||, concatenation; 0x, hexadecimal).

$$Adv_A^{SKKE}(l, N, q_H, q_{H'}) = \max\left\{\Pr\left[\left|Exp_A^{ind}(l, N, q_H, q_{H'})\right| = 1\right] - \frac{1}{2}, \Pr\left[Exp_A^{auth}(l, N, q_H, q_{H'}) = 1\right]\right\}. \quad (5)$$

5. Security Proof

In this section, firstly, we prove the confidentiality (namely, indistinguishable security) and authentication of the SKKE protocol, namely, Theorem 1 and Theorem 2. In the random Oracle model, we reduce the confidentiality to the collision of hash function and HMAC function and the indistinguishability between the output of random Oracle and random number, while the authentication is reduced to the collision of the hash function and the HMAC function. Then, we show the SKKE protocol is secure by Theorem 3.

Theorem 1. *For any PPT adversary A against the indistinguishability experiment, the advantage of the adversary A wins the experiment which satisfies the following inequality:*

$$Adv_{\mathscr{A}}^{ind}(l, N, q_H, q_{H'}) \le \frac{1}{2^{keylen}} + \frac{q_H}{2^{hashklen}} + \frac{q_{H'}}{2^{macklen}} + negl(hashklen) + negl(macklen), \quad (6)$$

where keylen *is the length of master key,* hashklen *and* macklen *are the output lengths of the hash function and the HMAC function,* q_H *and* $q_{H'}$ *are the times of the adversary queries to the hash functions* H *and* H', negl(hashklen) *and* negl(macklen) *are two negligible functions.*

Proof. For each game G_i, we denote $\mathrm{Exp}_A^{\mathrm{ind}}(l, N, q_H, q_{H'}) = 1$ as S_i. The initial game G_0 is defined as Figure 3.

Game G_1: the transition from G_0 to G_1 is realized by inlining the calculation process of LK_i and the HMAC function, as shown in Figure 5. So, the two games are equivalent. Then, we have

$$\Pr[S_1] = \Pr[S_0]. \quad (7)$$

Game G_2: if the adversary obtains the MK shared by the initiator and the responder, the adversary can calculate the LK, and then the adversary can win the game. Thus, as shown in Figure 6, in game G_2, we raise a flag bad_1 when the adversary obtains the MK by guessing. The difference in the probability of $(b = b')$ in this game and in game G_1 is bounded by the probability of bad_1 in the latter game, while the probability of bad_1 does not exceed $1/2^{keylen}$. Then, we have

$$\left|\Pr[S_2] - \Pr[S_1]\right| \le \Pr[G_2: bad_1] \le \frac{1}{2^{keylen}}. \quad (8)$$

Game G_3: as shown in Figure 7, in this game, we raise a flag bad_2 to represent that the adversary finds a collision of the outer hash function H' of the HMAC function $H'((r_i \oplus opad) \| H't(r_i \oplus ipad \| Data_i))$, then the adversary $\mathscr{A}$ obtains Z_i. In the RO model, the hash function is idealized as a random Oracle, and an Oracle $H_{\mathscr{A}}'$ is randomly selected from the function cluster $F^{x,y}$: $x \mapsto y$ to replace the hash function H'. Then, according to the lemma, that is, the output of the random Oracle which is indistinguishable from the random number, we randomly and evenly select λ_i and replace Z_i with λ_i. Therefore, the upper bound of probability difference of G_2 to G_3 transformation is characterized by the

$\text{Exp}_A^{\text{ind}}(l, N, q_H, q_{H'})$:	Oracles:
$(U_i, V_i) \leftarrow A_1()$;	Send(U_i, invoke):
$(MK_i, LK_i) \leftarrow ses(U_i, V_i)$;	$QEU_i \leftarrow \{0,1\}^l$;
$b \xleftarrow{\$} \{0,1\}^l$;	return (QEU_i)
$LK_i' \xleftarrow{\$} \{0,1\}^{keylen}$;	
$b' \leftarrow A_2(b ? LK_i, LK_i')$;	Send(V_i, QEU_i):
return $(b = b')$	$QEV_i \leftarrow \{0,1\}^l$;
	$Z_i \leftarrow MAC\{U \Vert V \Vert QEU_i \Vert QEV_i\}_{MK_i}$;
	$MacKey_i \leftarrow H(Z_i \Vert 0\text{x}00000001)$;
	$MacData_{1_i} \leftarrow 0\text{x}03 \Vert V \Vert U \Vert QEV_i \Vert QEU_i$;
	$MacTag_{1_i} \leftarrow MAC(MacData_{1_i})_{MacKey_i}$;
	return $(QEV_i, MacTag_{1_i})$
	Send$(U_i, (QEV_i, MacTag_{1_i}))$:
	$Z_i \leftarrow MAC\{U \Vert V \Vert QEU_i \Vert QEV_i\}_{MK_i}$;
	$MacKey_i \leftarrow H(Z_i \Vert 0\text{x}00000001)$;
	$MacData_{2_i} \leftarrow 0\text{x}02 \Vert U \Vert V \Vert QEU_i \Vert QEV_i$;
	$MacTag_{2_i} \leftarrow MAC(MacData_{2_i})_{MacKey_i}$;
	$MacData_{1_i} \leftarrow 0\text{x}03 \Vert V \Vert U \Vert QEV_i \Vert QEU_i$;
	if $(MAC(MacData_{1_i})_{MacKey_i} = MacTag_{1_i})$
	$ses(U_i, V_i) \leftarrow (MK_i, LK_i)$;
	return $(MacTag_{2_i})$
	else return $\perp$
	Send$(V_i, MacTag_{2_i})$:
	$MacData_{2_i} \leftarrow 0\text{x}02 \Vert U \Vert V \Vert QEU_i \Vert QEV_i$;
	if $(MAC(MacData_{2_i})_{MacKey_i} = MacTag_{2_i})$;
	$ses(U_i, V_i) \leftarrow (MK_i, LK_i)$;
	else return $\perp$
	Reveal(U_i, V_i):
	return (LK_i)
	Corrupt(U_i, V_i):
	return (MK_i)

Figure 3: Indistinguishability experiment of the SKKE protocol.

$\text{Exp}_A^{\text{auth-uv}}(l, N, q_H, q_{H'})$:	Oracles:
$((U_i, V_i), MK_i') \leftarrow A_1()$;	Send(V_i, QEU_i, MK_i'):
$(MK_i, LK_i) \leftarrow ses(U_i, V_i)$;	$QEV_i' \xleftarrow{\$} \{0,1\}^l$;
$QEU_i \xleftarrow{\$} \{0,1\}^l$;	$Z_i' \leftarrow MAC\{U \Vert V \Vert QEU_i \Vert QEV_i'\}_{MK_i'}$;
$(QEV_i', MacTag_{1_i}') \leftarrow A_2(QEU_i, MK_i')$;	$MacKey_i' \leftarrow H(Z_i' \Vert 0\text{x}00000001)$;
$Z_i \leftarrow MAC\{U \Vert V \Vert QEU_i \Vert QEV_i'\}_{MK_i}$;	$MacData_{1_i} \leftarrow 0\text{x}03 \Vert V \Vert U \Vert QEV_i' \Vert QEU_i$;
$MacKey_i \leftarrow H(Z_i \Vert 0\text{x}00000001)$;	$MacTag_{1_i}' \leftarrow MAC(MacData_{1_i})_{MacKey_i'}$;
$MacData_{1_i} \leftarrow 0\text{x}03 \Vert V \Vert U \Vert QEV_i' \Vert QEU_i$;	return $(QEV_i', MacTag_{1_i}')$
$MacTag_{1_i} \leftarrow MAC(MacData_{1_i})_{MacKey_i}$;	Reveal(U_i, V_i):
return $(MacTag_{1_i} = MacTag_{1_i}')$	return (LK_i)
	Corrupt(U_i, V_i):
	return (MK_i)

Figure 4: Authentication experiment of the SKKE protocol.

Game G_1:

$(U_i, V_i) \leftarrow A_1()$;

$(MK_i, LK_i) \leftarrow ses(U_i, V_i)$;

$b \xleftarrow{\$} \{0,1\}^l$; $LK_i' \xleftarrow{\$} \{0,1\}^{keylen}$;

$Data_i \leftarrow U \| V \| QEU_i \| QEV_i$;

$b' \leftarrow A_2(b ? H(H'((MK_i \oplus opad) \| H'((MK_i \oplus ipad) \| Data_i)) \| 0\text{x}00000002), LK_i')$;

return $(b = b')$

Oracles:

Send(U_i, invoke):

$QEU_i \leftarrow \{0,1\}^l$;

return (QEU_i)

Send(V_i, QEU_i):

$QEV_i \leftarrow \{0,1\}^l$; $Data_i \leftarrow U \| V \| QEU_i \| QEV_i$;

$Z_i \leftarrow H'((MK_i \oplus opad) \| H'(MK_i \oplus ipad) \| Data_i))$;

$MacKey_i \leftarrow H(Z_i \| 0\text{x}00000001)$;

$MacData_{1_i} \leftarrow 0\text{x}03 \| V \| U \| QEV_i \| QEU_i$;

$MacTag_{1_i} \leftarrow H'((MacKey_i \oplus opad) \| H'(MacKey_i \oplus ipad) \| MacData_{1_i}))$;

return $(QEV_i, MacTag_{1_i})$

Send$(U_i, (QEV_i, MacTag_{1_i}))$:

$Z_i \leftarrow H'((MK_i \oplus opad) \| H'(MK_i \oplus ipad) \| Data_i))$;

$MacKey_i \leftarrow H(Z_i \| 0\text{x}00000001)$;

$MacData_{2_i} \leftarrow 0\text{x}02 \| U \| V \| QEU_i \| QEV_i$;

$MacTag_{2_i} \leftarrow H'((MacKey_i \oplus opad) \| H'(MacKey_i \oplus ipad) \| MacData_{2_i}))$;

$MacData_{1_i} \leftarrow 0\text{x}03 \| V \| U \| QEV_i \| QEU_i$;

if $(H'((MacKey_i \oplus opad) \| H'(MacKey_i \oplus ipad) \| MacData_{1_i})) = MacTag_{1_i})$

$ses(U_i, V_i) \leftarrow (MK_i, H(Z_i \| 0\text{x}00000001))$;

return $(MacTag_{2_i})$

else return $\perp$

Send$(V_i, MacTag_{2_i})$:

$MacData_{2_i} \leftarrow 0\text{x}02 \| U \| V \| QEU_i \| QEV_i$;

if $(MAC(MacData_{2_i})_{MacKey_i} = MacTag_{2_i})$;

$ses(U_i, V_i) \leftarrow (MK_i, H(Z_i \| 0\text{x}00000001))$;

else return $\perp$

Reveal(U_i, V_i):

return (LK_i)

Corrupt(U_i, V_i):

return (MK_i)

Figure 5: Game G_1 of indistinguishability security of the SKKE protocol.

probability of bad_2 event occurrence and the indistinguishability between the output of the random Oracle and a random number. Then, we can get

$$\left|\Pr[S_3] - \Pr[S_2]\right| \leq \Pr[G_3 : bad_2] + negl(macklen) \leq \frac{q_{H'}}{2^{macklen}} + negl(macklen). \tag{9}$$

Game G_2 :

$(U_i, V_i) \leftarrow A_1()$;

$(MK_i, LK_i) \leftarrow ses(U_i, V_i)$;

$b \xleftarrow{\$} \{0,1\}^l$; $LK_i' \xleftarrow{\$} \{0,1\}^{keylen}$;

$Data_i \leftarrow U \| V \| QEU_i \| QEV_i$; $r_i \xleftarrow{\$} \{0,1\}^{keylen}$;

if $(r_i = MK_i)$ then

$bad_1 \leftarrow$ true ;

else

$b' \leftarrow A_2(b ? H(H'((r_i \oplus opad) \| H'((r_i \oplus ipad) \| Data_i)) \| 0x00000002), LK_i')$;

return $(b = b')$

Oracles :

Send$(U_i$, invoke$)$:

$QEU_i \leftarrow \{0,1\}^l$;

return (QEU_i)

Send(V_i, QEU_i) :

$QEV_i \leftarrow \{0,1\}^l$; $Data_i \leftarrow U \| V \| QEU_i \| QEV_i$;

$Z_i \leftarrow H'((r_i \oplus opad) \| H(r_i \oplus ipad) \| Data_i))$;

$MacKey_i \leftarrow H(Z_i \| 0x00000001)$;

$MacData_{1_i} \leftarrow 0x03 \| V \| U \| QEV_i \| QEU_i$;

$MacTag_{1_i} \leftarrow H'((MacKey_i \oplus opad) \| H'(MacKey_i \oplus ipad) \| MacData_{1_i}))$;

return $(QEV_i, MacTag_{1_i})$

Send$(U_i, (QEV_i, MacTag_{1_i}))$:

$Z_i \leftarrow H'((r_i \oplus opad) \| H(r_i \oplus ipad) \| Data_i))$;

$MacKey_i \leftarrow H(Z_i \| 0x00000001)$;

$MacData_{2_i} \leftarrow 0x02 \| U \| V \| QEU_i \| QEV_i$;

$MacTag_{2_i} \leftarrow H'((MacKey_i \oplus opad) \| H'(MacKey_i \oplus ipad) \| MacData_{2_i}))$;

$MacData_{1_i} \leftarrow 0x03 \| V \| U \| QEV_i \| QEU_i$;

if $(H'((MacKey_i \oplus opad) \| H'(MacKey_i \oplus ipad) \| MacData_{1_i})) = MacTag_{1_i})$

$ses(U_i, V_i) \leftarrow (r_i, H(Z_i \| 0x00000001))$;

return $(MacTag_{2_i})$

else return $\perp$

Send$(V_i, MacTag_{2_i})$:

$MacData_{2_i} \leftarrow 0x02 \| U \| V \| QEU_i \| QEV_i$;

if $(H'((MacKey_i \oplus opad) \| H'(MacKey_i \oplus ipad) \| MacData_{2_i})) = MacTag_{2_i})$;

$ses(U_i, V_i) \leftarrow (r_i, H(Z_i \| 0x00000001))$;

else return $\perp$

Reveal(U_i, V_i) :

return (LK_i)

Corrupt(U_i, V_i) :

return (MK_i)

Figure 6: Game G_2 of indistinguishability security of the SKKE protocol.

Game G_4: as shown in Figure 8, in this game G_4, we raise a flag bad_3 to represent that the adversary finds a collision of the hash function H, then the adversary $\mathcal{A}$ obtains LK_i. In the RO model, the hash function is idealized as a random Oracle, and an Oracle $H_{\mathcal{A}}$ is randomly selected from the function cluster $F^{x,y}$: $x \longmapsto y$ to replace the hash function H. Then, according to the lemma, that is, the output of the random Oracle which is indistinguishable from the random number, we randomly select γ_i and replace $H_{\mathcal{A}}(\lambda_i \| 0x00000002)$ with λ_i. Therefore, the upper bound of

Game G_3 :

$(U_i, V_i) \leftarrow A_1()$; $(MK_i, LK_i) \leftarrow ses(U_i, V_i)$;

$b \xleftarrow{\$} \{0,1\}^l$; $LK_i' \xleftarrow{\$} \{0,1\}^{keylen}$;

$Data_i \leftarrow U \| V \| QEU_i \| QEV_i$; $r_i \xleftarrow{\$} \{0,1\}^{keylen}$;

if $(r_i = MK_i)$ then

$\text{bad}_1 \leftarrow \text{true}$;

else

$\lambda_i \xleftarrow{\$} \{0,1\}^{macklen}$;

if $(\lambda_i = H_A'((r_i \oplus opad) \| H_A'((r_i \oplus ipad) \| Data_i)))$ then

$\text{bad}_2 \leftarrow \text{true}$;

else

$b' \leftarrow A_2(b ? H(\lambda_i \| 0\text{x}00000002), LK_i')$;

return $(b = b')$

Oracles :

Send(U_i, invoke) :

$QEU_i \leftarrow \{0,1\}^l$;

return (QEU_i)

Send(V_i, QEU_i) :

$QEV_i \leftarrow \{0,1\}^l$;

$MacKey_i \leftarrow H(\lambda_i \| 0\text{x}00000001)$;

$MacData_{1_i} \leftarrow 0\text{x}03 \| V \| U \| QEV_i \| QEU_i$;

$MacTag_{1_i} \leftarrow H_A'((MacKey_i \oplus opad) \| H_A'(MacKey_i \oplus ipad) \| MacData_{1_i}))$;

return $(QEV_i, MacTag_{1_i})$

Send$(U_i, (QEV_i, MacTag_{1_i}))$:

$MacKey_i \leftarrow H(\lambda_i \| 0\text{x}00000001)$;

$MacData_{2_i} \leftarrow 0\text{x}02 \| U \| V \| QEU_i \| QEV_i$;

$MacTag_{2_i} \leftarrow H_A'((MacKey_i \oplus opad) \| H_A'(MacKey_i \oplus ipad) \| MacData_{2_i}))$;

$MacData_{1_i} \leftarrow 0\text{x}03 \| V \| U \| QEV_i \| QEU_i$;

if $(H_A'((MacKey_i \oplus opad) \| H_A'(MacKey_i \oplus ipad) \| MacData_{1_i})) = MacTag_{1_i})$

$ses(U_i, V_i) \leftarrow (r_i, H(\lambda_i \| 0\text{x}00000001))$;

return $(MacTag_{2_i})$

else return $\perp$

Send$(V_i, MacTag_{2_i})$:

$MacData_{2_i} \leftarrow 0\text{x}02 \| U \| V \| QEU_i \| QEV_i$;

if $(H_A'((MacKey_i \oplus opad) \| H_A'(MacKey_i \oplus ipad) \| MacData_{2_i})) = MacTag_{2_i})$;

$ses(U_i, V_i) \leftarrow (r_i, H(\lambda_i \| 0\text{x}00000001))$;

else return $\perp$

Reveal(U_i, V_i) :

return (LK_i)

Corrupt(U_i, V_i) :

return (MK_i)

Figure 7: Game G_3 of indistinguishability security of the SKKE protocol.

Game G_4 :

$(U_i, V_i) \leftarrow A_1()$;

$(MK_i, LK_i) \leftarrow ses(U_i, V_i)$;

$b \xleftarrow{\$} \{0,1\}^l$; $LK_i' \xleftarrow{\$} \{0,1\}^{keylen}$;

$Data_i \leftarrow U \| V \| QEU_i \| QEV_i$; $r_i \xleftarrow{\$} \{0,1\}^{keylen}$;

if $(r_i = MK_i)$ then

$\text{bad}_1 \leftarrow \text{true}$;

else

$\lambda_i \xleftarrow{\$} \{0,1\}^{macklen}$;

if $(\lambda_i = H_A'((r_i \oplus opad) \| H_A'((r_i \oplus ipad) \| Data_i)))$ then

$\text{bad}_2 \leftarrow \text{true}$;

else

$\gamma_i \xleftarrow{\$} \{0,1\}^{hashklen}$;

if $(\gamma_i = H(\lambda_i \| 0\text{x}00000002))$ then

$\text{bad}_3 \leftarrow \text{true}$;

else

$b' \leftarrow A_2(b ? \gamma_i, LK_i')$;

return $(b = b')$

Oracles :

Send(U_i, invoke) :

$QEU_i \leftarrow \{0,1\}^l$;

return (QEU_i)

Send(V_i, QEU_i) :

$QEV_i \leftarrow \{0,1\}^l$;

$MacKey_i \leftarrow H(\lambda_i \| 0\text{x}00000001)$;

$MacData_{1_i} \leftarrow 0\text{x}03 \| V \| U \| QEV_i \| QEU_i$;

$MacTag_{1_i} \leftarrow H_A'((MacKey_i \oplus opad) \| H_A'(MacKey_i \oplus ipad) \| MacData_{1_i}))$;

return $(QEV_i, MacTag_{1_i})$

Send$(U_i, (QEV_i, MacTag_{1_i}))$:

$MacKey_i \leftarrow H(\lambda_i \| 0\text{x}00000001)$;

$MacData_{2_i} \leftarrow 0\text{x}02 \| U \| V \| QEU_i \| QEV_i$;

$MacTag_{2_i} \leftarrow H_A'((MacKey_i \oplus opad) \| H_A'(MacKey_i \oplus ipad) \| MacData_{2_i}))$;

$MacData_{1_i} \leftarrow 0\text{x}03 \| V \| U \| QEV_i \| QEU_i$;

if $(H_A'((MacKey_i \oplus opad) \| H_A'(MacKey_i \oplus ipad) \| MacData_{1_i})) = MacTag_{1_i})$

$ses(U_i, V_i) \leftarrow (r_i, \gamma_i)$;

return $(MacTag_{2_i})$

else return $\perp$

Send$(V_i, MacTag_{2_i})$:

$MacData_{2_i} \leftarrow 0\text{x}02 \| U \| V \| QEU_i \| QEV_i$;

if $(H_A'((MacKey_i \oplus opad) \| H_A'(MacKey_i \oplus ipad) \| MacData_{2_i})) = MacTag_{2_i})$;

$ses(U_i, V_i) \leftarrow (r_i, \gamma_i)$;

else return $\perp$

Reveal(U_i, V_i) :

return (LK_i)

Corrupt(U_i, V_i) :

return (MK_i)

Figure 8: Game G_4 of indistinguishability security of the SKKE protocol.

the probability difference of G_3 to G_4 transformation is characterized by the probability of bad_3 event occurrence and the indistinguishability between the output of the random Oracle and a random number. Then, we have

$$\left|\Pr[S_4] - \Pr[S_3]\right| \le \Pr[G_4: bad_3] + negl(hashklen) \le \frac{q_H}{2^{hashklen}} + negl(hashklen), \tag{10}$$

where hashklen is the output length of the hash function and negl(hashklen) is a negligible function. While in G_4, the protocol copies returned are random numbers that have nothing to do with coin tossing, then we have

$$\Pr[S_4] = \frac{1}{2}. \tag{11}$$

In summary, from (1) to (5), we can get the adversary's advantage in winning as follows:

$$Adv_{\mathscr{A}}^{ind}(l, N, q_H, q_{H'}) \le \frac{1}{2^{keylen}} + \frac{q_H}{2^{hashklen}} + \frac{q_{H'}}{2^{macklen}} + negl(hashklen) + negl(macklen). \tag{12}$$

Theorem 2. *For any PPT adversary $\mathscr{A}$ against the authentication experiment, the advantage of the adversary winning satisfies the following inequality:*

$$Adv_{\mathscr{A}}^{auth}(l, N, q_H, q_{H'}) \le \frac{1}{2^{keylen-1}} + \frac{q_H}{2^{hashklen-1}} + \frac{q_{H'}}{2^{macklen-1}}, \tag{13}$$

where keylen *is the length of master key;* hashklen *and* macklen *are the output lengths of the HMAC function and the hash function; q_H is the times of the adversary queries to the hash function H; and $q_{H'} = s_1 + s_2$ is the total times of the adversary queries to the hash function H'.*

Proof. For each game G_i, we denote $Exp_{\mathscr{A}}^{in\,d}(l, N, q_H, q_{H'}) = 1$ as S_i. The initial game G_0 is defined as Figure 4.

Game G_1: as shown in Figure 9, the transition from G_0 to G_1 is realized by in-lining the calculation process of HMAC function. Thus, the two games are equivalent. Then, we can have

$$\Pr[S_1] = \Pr[S_0]. \tag{14}$$

Game G_2: as shown in Figure 10, we raise a flag bad_1 to represent that the adversary finds a collision of the hash function of H', then the adversary $\mathscr{A}$ obtains a $MacKey_i'$, which satisfies the following conditions:

$$MacTag_{1_i} = MacTa'_{g1_i}, \tag{15}$$

$$(MacKey_i \oplus opad) \| H'(MacKey_i \oplus ipad) \| MacData_{1_i} \ne (MacKey_i' \oplus opad) \| H'(MacKey_i' \oplus ipad) \| MacData_{1_i}. \tag{16}$$

Since the probability of the bad_1 event happening is not more than $S_1/2^{macklen}$, then we can have

$$\left|\Pr[S_2] - \Pr[S_1]\right| \le \Pr[G_2: bad_1] \le \frac{s_1}{2^{macklen}}, \tag{17}$$

where s_1 is the number of times the adversary queries to H' in this game.

Game G_3: as shown in Figure 11, we raise a flag bad_2 to represent that the adversary finds a collision of the hash function H, then the adversary $\mathscr{A}$ obtains a Z_i', which satisfies the following conditions: $MacKey_i = MacKey_i'$ and $Z_i \ne Z_i'$. Since the probability of the bad_2 event happening is not more than $q_H/2^{hashklen}$, then we can have

$$\left|\Pr[S_3] - \Pr[S_2]\right| \le \Pr[G_3: bad_2] \le \frac{q_H}{2^{hashklen}}. \tag{18}$$

Game G_4: as shown in Figure 12, we raise a flag bad_3 to represent that the adversary finds a collision of the hash function of H', then the adversary $\mathscr{A}$ obtains a MK_i', which satisfies $Z_i = Z_i'$ and $MK_i \ne MK_i'$. Since the probability of the bad_3 event happening is not more than $s_2/2^{macklen}$, then we can have

$$\left|\Pr[S_4] - \Pr[S_3]\right| \le \Pr[G_4: bad_3] \le \frac{s_2}{2^{macklen}}, \tag{19}$$

where s_2 is the number of times the adversary queries to H' in this game. Due to the inclusion relationship between events $H'((MacKey_i \oplus opad) \| H'((MacKey_i \oplus ipad) \| MacData_{1_i})) = H'(MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i})$, $MacKey_i = MacKey_i'$, $Z_i = Z_i'$, and $MK_i = MK_i'$, $\Pr[S_4] = \Pr[G_4: MK_i = MK_i']$ can be obtained from the conditional probability formula. The probability of the adversary obtains the master key MK_i by guessing which is not more than $1/2^{keylen}$ ($keylen$ is the length of the master key), so we can have

Game G_1 :

$((U_i, V_i), MK_i') \leftarrow A_1()$;

$(MK_i, \_\_) \leftarrow ses(U_i, V_i)$;

$QEU_i \xleftarrow{\$} \{0,1\}^l$;

$(QEV_i', MacTag_{1_i}') \leftarrow A_2(QEU_i, MK_i')$;

$Data_i \leftarrow U \| V \| QEU_i \| QEV_i$;

$Z_i \leftarrow H'(MK_i \oplus opad) \| H'((MK_i \oplus ipad) \| Data_i))$;

$MacKey_i \leftarrow H(Z_i \| 0\text{x}00000001)$;

$MacData_{1_i} \leftarrow 0\text{x}03 \| V \| U \| QEV_i' \| QEU_i$;

return $(H'((MacKey_i \oplus opad) \| H'((MacKey_i \oplus ipad) \| MacData_{1_i}))$

$= H'(MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i})))$

Oracles :

Send(V_i, QEU_i, MK_i') :

$QEV_i' \xleftarrow{\$} \{0,1\}^l$; $Data_i \leftarrow U \| V \| QEU_i \| QEV_i$

$Z_i' \leftarrow H'(MK_i' \oplus opad) \| H'((MK_i' \oplus ipad) \| Data_i))$;

$MacKey_i' \leftarrow H(Z_i' \| 0\text{x}00000001)$; $MacData_{1_i} \leftarrow 0\text{x}03 \| V \| U \| QEV_i' \| QEU_i$;

$MacTag_{1_i}' \leftarrow H'(MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$;

return $(QEV_i', MacTag_{1_i}')$

Corrupt(U_i, V_i) :

return (MK_i)

Figure 9: Game G_1 of the authentication of the SKKE protocol.

Game G_2 :

$((U_i, V_i), MK_i') \leftarrow A_1()$; $(MK_i, LK_i) \leftarrow ses(U_i, V_i)$;

$QEU_i \xleftarrow{\$} \{0,1\}^l$; $(QEV_i', MacTag_{1_i}') \leftarrow A_2(QEU_i, MK_i')$;

$Z_i \leftarrow MAC\{U \| V \| QEU_i \| QEV_i'\}_{MK_i}$;

$MacKey_i \leftarrow H(Z_i \| 0\text{x}00000001)$;

$MacData_{1_i} \leftarrow 0\text{x}03 \| V \| U \| QEV_i' \| QEU_i$;

if $(H'((MacKey_i \oplus opad) \| H'((MacKey_i \oplus ipad) \| MacData_{1_i}))$

$= H'((MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$

$\Lambda((MacKey_i \oplus opad) \| H'((MacKey_i \oplus ipad) \| MacData_{1_i})$

$\neq MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$

then $\text{bad}_1 \leftarrow \text{true}$;

else

return $(H'((MacKey_i \oplus opad) \| H'((MacKey_i \oplus ipad) \| MacData_{1_i}))$

$= H'(MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$

$\Lambda(MacKey_i = MacKey_i'))$

Oracles :

Send(V_i, QEU_i, MK_i') :

$QEV_i' \xleftarrow{\$} \{0,1\}^l$; $Data_i \leftarrow U \| V \| QEU_i \| QEV_i$

$Z_i' \leftarrow H'(MK_i' \oplus opad) \| H'((MK_i' \oplus ipad) \| Data_i))$;

$MacKey_i' \leftarrow H(Z_i' \| 0\text{x}00000001)$; $MacData_{1_i} \leftarrow 0\text{x}03 \| V \| U \| QEV_i' \| QEU_i$;

$MacTag_{1_i}' \leftarrow H'(MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$;

return $(QEV_i', MacTag_{1_i}')$

Corrupt(U_i, V_i) :

return (MK_i)

Figure 10: Game G_2 of the authentication of the SKKE protocol.

Game G_3 :

$((U_i, V_i), MK_i') \leftarrow A_1()$; $(MK_i, LK_i) \leftarrow ses(U_i, V_i)$;

$QEU_i \xleftarrow{\$} \{0,1\}^l$; $(QEV_i', MacTag_{1_i}') \leftarrow A_2(QEU_i, MK_i')$;

$Z_i \leftarrow MAC\{U \| V \| QEU_i \| QEV_i'\}_{MK_i}$;

$MacKey_i \leftarrow H(Z_i \| 0\text{x}00000001)$;

$MacData_{1_i} \leftarrow 0\text{x}03 \| V \| U \| QEV_i' \| QEU_i$;

if $(H'((MacKey_i \oplus opad) \| H'((MacKey_i \oplus ipad) \| MacData_{1_i}))$

$= H'((MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$

$\Lambda((MacKey_i \oplus opad) \| H'((MacKey_i \oplus ipad) \| MacData_{1_i})$

$\neq MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$

then $\text{bad}_1 \leftarrow \text{true}$;

else

if $(H'((MacKey_i \oplus opad) \| H'((MacKey_i \oplus ipad) \| MacData_{1_i}))$

$= H'((MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$

$\Lambda(H(Z_i \| 0x00000001) = H(Z_i' \| 0x00000001))$

$\Lambda(Z_i \neq Z_i')$

then $\text{bad}_2 \leftarrow \text{true}$;

else

return $(H'((MacKey_i \oplus opad) \| H'((MacKey_i \oplus ipad) \| MacData_{1_i}))$

$= H'(MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$

$\Lambda(MacKey_i = MacKey_i')$

$\Lambda(Z_i = Z_i'))$

Oracles :

Send(V_i, QEU_i, MK_i') :

$QEV_i' \xleftarrow{\$} \{0,1\}^l$; $Data_i \leftarrow U \| V \| QEU_i \| QEV_i$

$Z_i' \leftarrow H'(MK_i' \oplus opad) \| H'((MK_i' \oplus ipad) \| Data_i))$;

$MacKey_i' \leftarrow H(Z_i' \| 0\text{x}00000001)$; $MacData_{1_i} \leftarrow 0\text{x}03 \| V \| U \| QEV_i' \| QEU_i$;

$MacTag_{1_i}' \leftarrow H'(MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$;

return $(QEV_i', MacTag_{1_i}')$

Corrupt(U_i, V_i) :

return (MK_i)

Figure 11: Game G_3 of the authentication of the SKKE protocol.

$$\left|\Pr[S_4] = \Pr[G_4 \colon MK_i' = MK_i]\right| \leq \frac{1}{2^{keylen}}. \tag{20}$$

To sum up, from (6) to (10), we can get the advantage of the adversary against the authentication experiment of U to V, and it satisfies the following inequality:

$$Adv_{\mathcal{A}}^{auth-uv}(l, N, q_H, q_{H'}) \leq \frac{1}{2^{keylen}} + \frac{q_H}{2^{hashklen}} + \frac{q_{H'}}{2^{macklen}}. \tag{21}$$

In a word, the advantage of the adversary against the authentication of the SKKE protocol satisfies

$$Adv_A^{auth}(l, N, q_H, q_{H'}) \leq \frac{1}{2^{keylen-1}} + \frac{q_H}{2^{hashklen-1}} + \frac{q_{H'}}{2^{macklen-1}}. \tag{22}$$

□

Theorem 3. *For any PPT adversary $\mathcal{A}$ against the security of the SKKE protocol, the advantage of the adversary winning satisfies the following inequality:*

$$Adv_{\mathcal{A}}^{SKKE}(l, N, q_H, q_{H'}) \leq \frac{1}{2^{keylen-1}} + \frac{q_H}{2^{hashklen-1}} + \frac{q_{H'}}{2^{macklen-1}} + negl(hashklen) + negl(macklen). \tag{23}$$

Game G_4 :

$((U_i, V_i), MK_i') \leftarrow A_1()$; $(MK_i, LK_i) \leftarrow ses(U_i, V_i)$;

$QEU_i \xleftarrow{\$} \{0,1\}^l$; $(QEV_i', MacTag_{1_i}') \leftarrow A_2(QEU_i, MK_i')$;

$Z_i \leftarrow MAC\{U \| V \| QEU_i \| QEV_i'\}_{MK_i}$;

$MacKey_i \leftarrow H(Z_i \| 0\text{x}00000001)$;

$MacData_{1_i} \leftarrow 0\text{x}03 \| V \| U \| QEV_i' \| QEU_i$;

if $(H'((MacKey_i \oplus opad) \| H'((MacKey_i \oplus ipad) \| MacData_{1_i}))$

$= H'((MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$

$\Lambda((MacKey_i \oplus opad) \| H'((MacKey_i \oplus ipad) \| MacData_{1_i})$

$\neq MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$

then $\text{bad}_1 \leftarrow \text{true}$;

else

if $(H'((MacKey_i \oplus opad) \| H'((MacKey_i \oplus ipad) \| MacData_{1_i}))$

$= H'((MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$

$\Lambda(H(Z_i \| 0x00000001) = H(Z_i' \| 0x00000001))$

$\Lambda(Z_i \neq Z_i')$

then $\text{bad}_2 \leftarrow \text{true}$;

else

if $(H'((MacKey_i \oplus opad) \| H'((MacKey_i \oplus ipad) \| MacData_{1_i}))$

$= H'((MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$

$\Lambda(H(Z_i \| 0x00000001) = H(Z_i' \| 0x00000001))$

$\Lambda(H'((MK_i \oplus opad) \| H'((MK_i \oplus ipad) \| Data_i))$

$= H'((MK_i' \oplus opad) \| H'((MK_i' \oplus ipad) \| Data_i))$

$\Lambda((MK_i \oplus opad) \| H'((MK_i \oplus ipad) \| Data_i)$

$\neq (MK_i \oplus opad) \| H'((MK_i \oplus ipad) \| Data_i))$

then $\text{bad}_3 \leftarrow \text{true}$;

else

return $(H'((MacKey_i \oplus opad) \| H'((MacKey_i \oplus ipad) \| MacData_{1_i}))$

$= H'(MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$

$\Lambda(MacKey_i = MacKey_i')$

$\Lambda(Z_i = Z_i')$

$\Lambda(MK_i = MK_i'))$

Oracles :

Send(V_i, QEU_i, MK_i') :

$QEV_i' \xleftarrow{\$} \{0,1\}^l$; $Data_i \leftarrow U \| V \| QEU_i \| QEV_i$

$Z_i' \leftarrow H'(MK_i' \oplus opad) \| H'((MK_i' \oplus ipad) \| Data_i))$;

$MacKey_i' \leftarrow H(Z_i' \| 0\text{x}00000001)$; $MacData_{1_i} \leftarrow 0\text{x}03 \| V \| U \| QEV_i' \| QEU_i$;

$MacTag_{1_i}' \leftarrow H'(MacKey_i' \oplus opad) \| H'((MacKey_i' \oplus ipad) \| MacData_{1_i}))$;

return $(QEV_i', MacTag_{1_i}')$

Corrupt(U_i, V_i) :

return (MK_i)

Figure 12: Game G_4 of the authentication of the SKKE protocol.

Remark 2. The proof of Theorem 3 is obvious. It should be noted that q_H is the largest times of the adversary queries to the hash function H in these two security experiments, and $q_{H'}$ is the largest total times of the adversary queries to the hash function H', similarly.

6. Conclusions

As an Internet of things communication technology, ZigBee has a wide range of applications in home automation, smart energy, commercial building automation, personal, home and hospital care, telecom, and wireless sensor. At the same time, its security has also received more and more attention. According to the standard of ZigBee 3.0, this paper considers the security of the SKKE protocol, which is used by the initiator and the responder to establish the application link key. In the random Oracle model, we reduce the confidentiality of the SKKE protocol to the collision of the hash function and the HMAC function and the indistinguishability between the output of random Oracle and random number. Then, we reduce the authentication to the collision between the hash function and the HMAC function. Finally, we give the theoretical proof of the confidentiality and authentication of the SKKE protocol with the game-based method. As we known, there is a lack of research on the provable security of the ZigBee protocol at this stage, so it is a meaningful work to study the provable security of the SKKE protocol, and it is helpful to promote further research of the ZigBee protocol security.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] C. Ge, Z. Liu, J. Xia, and L. Fang, "Revocable identity-based broadcast proxy Re-encryption for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1214–1226, 2021.

[2] C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski, and F. Liming, "Secure keyword search and data sharing mechanism for cloud computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 6, p. 1, 2020.

[3] J. Durech and M. Franekova, "Security attacks to ZigBee technology and their practical realization," *2014 IEEE 12th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, IEEE, in *Proceedings of the 2014 IEEE 12th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pp. 345–349, January 2014.

[4] L. N. Whitehurst, T. R. Andel, and J. T. McDonald, "Exploring security in ZigBee networks," *Proceedings of the 9th Annual Cyber and Information Security Research Conference on - CISR '14*, ACMDL, in *Proceedings of the 9th Annual Cyber and Information Security Research Conference*, pp. 25–28, April 2014.

[5] X. Cao, D. M. Shila, Z. Yang, Y. Zhou, and J. Chen, "Ghost-in-Zigbee: Energy depletion attack on zigbee-based wireless networks," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 816–829, 2016.

[6] A. Gawanmeh, "Embedding and verification of ZigBee protocol stack in event-B," *Procedia Computer Science*, vol. 5, pp. 736–741, 2011.

[7] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "A verifiable and fair attribute-based proxy Re-encryption scheme for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2021.

[8] R. M. Nadeem and A. A. Gill, "A formal model for verification of ZigBee protocol for secure network authentication," *Indian Journal of Science and Technology*, vol. 10, no. 20, pp. 1–7, 2017.

[9] A. P. Melaragno, D. Bandara, D. Wijesekera, and J. B. Michael, "Securing the ZigBee protocol in the Smart grid," *Computer*, vol. 45, no. 4, pp. 92–94, 2012.

[10] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "Revocable attribute-based encryption with data integrity in clouds," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2021.

[11] M. Alshahrani, I. Traore, and I. Woungang, "Anonymous mutual IoT interdevice authentication and key agreement scheme based on the ZigBee technique," *Internet of Things*, vol. 7, p. 100061, Article ID 100061, 2019.

[12] K. Hofer-Schmitz and B. Stojanović, "Towards formal verification of IoT protocols: a Review," *Computer Networks*, vol. 174, p. 107233, 2020.

[13] L. Li, P. Podder, and E. Hoque, "A formal security analysis of ZigBee (1.0 and 3.0)," *Proceedings of the 7th Symposium on Hot Topics in the Science of Security*, ACMDL, in *Proceedings of the 7th Symposium on Hot Topics in the Science of Security*, pp. 1–11, September 2020.

[14] B. Blanchet, "Mechanizing game-based proofs of security protocols," *Software Safety and Security*, vol. 33, pp. 1–25, 2012.

[15] E. Yuksel, H. R. Nielson, and F. Nielson, "ZigBee-2007 security essentials," in *Proceedings of the 13th Nordic Workshop on Secure IT Systems*, pp. 65–82, Technical University of Denmark, Kongens Lyngby, Denmark, October 2008.

[16] *ZigBee Specifications*, 2015, https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf.

[17] M. Bellare and P. Rogaway, "Random oracles are practical," *Proceedings of the 1st ACM conference on Computer and communications security - CCS '93*, Computer and Communications Security, in *Proceedings of the 1st ACM conference on Computer and communications security - CCS '93*, pp. 62–73, December 1993.

[18] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: keyed-hashing for message authentication," *RFC*, vol. 2104, pp. 1–11, 1997.