Hindawi
Mobile Information Systems
Volume 2021, Article ID 7314508, 18 pages
https://doi.org/10.1155/2021/7314508

Research Article

Scrutinizing the Vulnerability of Ephemeral Diffie–Hellman over COSE (EDHOC) for IoT Environment Using Formal Approaches

Jiyoon Kim,[1] **Daniel Gerbi Duguma**,[1] **Sangmin Lee**,[1] **Bonam Kim**,[1] **JaeDeok Lim**,[2] **and Ilsun You**[1]

[1] *Dept. of Information Security Engineering, Soonchunhyang University, Asan 31538, Republic of Korea*
[2] *Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, Republic of Korea*

Correspondence should be addressed to Ilsun You; ilsunu@gmail.com

Received 9 July 2021; Revised 4 August 2021; Accepted 16 August 2021; Published 13 September 2021

Academic Editor: Zengpeng Li

Most existing conventional security mechanisms are insufficient, mainly attributable to their requirements for heavy processing capacity, large protocol message size, and longer round trips, for resource-intensive devices operating in an Internet of Things (IoT) context. These devices necessitate efficient communication and security protocols that are cognizant of the severe resource restrictions regarding energy, computation, communication, and storage. To realize this, the IETF (Internet Engineering Task Force) is currently working towards standardizing an ephemeral key-based lightweight and authenticated key exchange protocol called EDHOC (Ephemeral Diffie–Hellman over COSE). The protocol's primary purpose is to build an OSCORE (Object Security for Constrained RESTful Environments) security environment by supplying crucial security properties such as secure key exchange, mutual authentication, perfect forward secrecy, and identity protection. EDHOC will most likely dominate IoT security once it becomes a standard. It is, therefore, imperative to inspect the protocol for any security flaw. In this regard, two previous studies have shown different security vulnerabilities of the protocol using formal security verification methods. Yet, both missed the vital security flaws we found in this paper: resource exhaustion and privacy attacks. In finding these vulnerabilities, we leveraged BAN-Logic and AVISPA to formally verify both EDHOC protocol variants. Consequently, we described these security flaws together with the results of the related studies and put forward recommended solutions as part of our future work.

1. Introduction

IoT refers to a network environment in which all surrounding objects are connected to wired and wireless networks to interact and exchange information over the Internet. These objects (also referred to as "things") can range from a simple soil moisture sensor in a field to a complex implanted device in a human body. With continuous developments in low-cost electronics (such as sensors), fast progress in mobile communication (especially with the introduction of 5G), and significant advances in data analytics (e.g., machine learning and lightweight deep learning), IoT has become one of the most demanded technologies in our time [1, 2]. Currently, IoT serves as an instrumental platform to host many applications in manufacturing, healthcare, energy, cities, and many more. In the next four years (by 2025) only, the total market share of IoT can stretch to reach up to 3 trillion USD [3], while the number of devices operating in an IoT environment can cross 42 billion with over 73 ZB of generated data [4].

Despite the vast expansion of IoT-enabled devices and their widespread applications, IoT still has several challenges that needs to be tackled. Some of the issues are tightly related to the severe computing resource constraints concerning storage, processing, and communication [5–7]. Such tight requirements call for efficient mechanisms to enable devices operating within IoT environments to function through unstable channels with constrained bandwidth and varying topology [8]. To realize these stringent conditions, essential protocols, such as [9–11], have been standardized by IETF. In addition, because IoT devices transport several sensitive information, security problems can threaten the inability to provide services and the user's personal information. Some potential security attacks are device software malfunction, prying, malevolent code infusions, device tampering, and unauthorized access [12]. Furthermore, studies such as [13, 14] investigated the security issues of integrating LPWAN in the 5G ecosystem, as well as the practical evaluation of compression and fragmentation of standard protocols as applied to IoTs in LPWAN, respectively. Hence, IoT devices require more capable security schemes that work in tandem with the communication protocols to mitigate these security attacks.

Even though IoT applications anticipate solid security assurance, securing IoT frameworks is challenging. It is mainly because of their intrinsic nature of resource constraint and absence of "security aware" design. Although there are various security solutions designed for conventional networks, such as IKEv2 [15], TLS [16], and DTLS [17], they are not suitable for the IoT environment due to their high degree of processing power and memory space. For instance, the footprint in bytes for a DTLS is six times heavier than the EDHOC + CoAP (Constrained Application Protocol) [8]. Fortunately, there are now efforts in designing standard security protocols mainly intended to serve in IoT environments. One such application layer security protocol is the OSCORE [18]. The protocol is efficient for severely constrained networks as it maintains the minor communication overhead possible. Using OSCORE, however, requires preshared keys to establish a security context. For this purpose, the IETF is in the process of standardizing an authenticated Diffie–Hellman key exchange protocol known as EDHOC [19]. The protocol is aforesaid to provide essential security properties such as mutual authentication, perfect forward secrecy, identity protection, and cipher suite negotiation.

EDHOC will most certainly dominate IoT security once it becomes a standard, which is why it is critical to analyse it for security vulnerabilities thoroughly. Since its inception in March 2016, it passed through 26 different versions, among which only two of its versions ([20] in 2018 and [21] in 2020) were formally analysed by [22, 23] using ProVerif [24] and Tamarin [25], respectively. While these studies bring numerous essential security issues to light, there are still security flaws that they have not yet discovered. Furthermore, evaluating security protocols using several formal approaches increases our confidence in the protocols' resilience to various security threats since one can compensate for the weakness of the other. Accordingly, in this paper, we formally analysed both the symmetric and asymmetric variants of the EDHOC protocol by using BAN (Burrows, Abadi, and Needham)-Logic [26] and AVISPA (Automated Validation of Internet Security Protocols and Applications) [27] to uncover other security issues. The formal verification results indicate that the protocols suffer from resource exhaustion and privacy attacks. While the former vulnerability is related to a class of attacks known as (distributed) denial-of-service attacks, where excessive and unnecessary requests deplete a node's resource, the latter pertains to privacy violations due to *ID_PSK* and ID_CRED_R (in symmetric and asymmetric variations, respectively). Both security issues are described in detail in Section 4 of the paper. The core contributions of this paper are summarized as follows:

(i) We carried out a formal security verification of the asymmetric and symmetric variants of the EDHOC protocol using two formal approaches: BAN-Logic and AVISPA

(ii) We pointed out two novel potential security vulnerabilities that may lead to resource exhaustion and privacy attacks

(iii) We described a concise summary of the principal security threats found by former related studies together with those identified by us

The remainder of the paper is organized as follows. Section 2 describes the EDHOC protocol along with the related studies on its formal security analysis. The formal verification of the protocol and results, respectively, are presented in the subsequent two sections. Finally, Section 5 concludes the paper.

2. The EDHOC Protocol

2.1. Protocol Overview. The increasing usage of IoT devices in vertical applications, such as energy, smart factory, healthcare, and transportation, calls for more efficient approaches to power, communication, storage, and processing. Given their severe constraint concerning these requirements, it is not possible (or inefficient) to apply existing security protocols. The main reason is due to the heavy cryptography algorithms, message sizes, and total round trips involved with these schemes. Implementing security on the application layer of the IoT communication systems is especially beneficial when there is insufficient security at the transport layer or when considering the performance of the communication is required. To this point, there are fundamental advances in providing application-aware security solutions. Some of these schemes are the CoAP [9] and its lightweight extension to provide sufficient object security, OSCORE [18].

Another vital protocol that serves as a lightweight authenticated key exchange mechanism for OSCORE is the EDHOC. The EDHOC protocol provides session key establishment while supporting fundamental security properties like perfect forward secrecy and mutual authentication [19]. The protocol involves essential components like Elliptic Curve Diffie–Hellman (ECDH) for key exchange, CBOR (Concise Binary Object Representation) [10] for data encoding, COSE (CBOR Object Signing and Encryption) [28] for protecting the CBOR encoded messages, and CoAP for message transportation. In summary, the primary intent of EDHOC is to leverage the OSCORE initiated security so that the message footprints and the round trips are small. Figure 1 shows the IoT protocol stack with the EDHOC protocol located in the application layer.

2.2. Related Works. Formal security analysis of various authentication protocols has been performed to guarantee the resilience of different security schemes against numerous

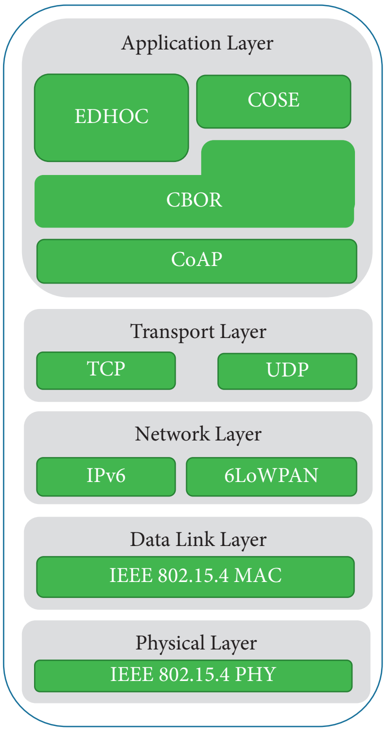


Figure 1: IoT protocol stack.

attacks. Concerning EDHOC, there are two significant studies that analysed the security of this protocol with a formal approach.

In [22], the authors formally analysed both symmetric-key and asymmetric-key options of the EDHOC protocol using ProVerif [24]. This research inspected the protocol against various security characteristics like identity protection against an active attacker, application data confidentiality and perfect forward secrecy, and robust authentication. Consequently, the authors highlighted the risk of leaking the responder's identity, although the initiator's identity is secured. Furthermore, by utilizing the same preshared key identifier ID_PSK, an attacker may link several sessions and launch various assaults to the symmetric variant of the protocol. Concerning the application data (AD_1 to AD_3), the authors also showed that only AD_3 (for both symmetric and asymmetric variants) satisfies secrecy, perfect forward secrecy, and integrity at both the time of message arrival and conclusion of the protocol.

Another paper [23] that analysed the EDHOC security using the Tamarin prover [25] verification tool found various improvement points. The authors, among other issues, identified the following flaws by the time they analysed the protocol: absence of nonrepudiation security property and lack of verification of ID_CRED_R of Msg2 by the initiator. The authors also showed that a security threat due to a prolonged metasession covering several sessions of the EDHOC protocol can happen when the responder rejects proposed cipher suites. The paper also recommended the use of a trusted execution environment (TEE) for security hardening.

2.3. Protocol Description. The initiator and responder of the EDHOC protocol can encrypt and protect the integrity of information communicated between them by following a similar construction as SIGMA-1 [29]. The initiator and responder exchange three messages to establish Diffie–Hellman's shared secrets and perform encryption using Authenticated Encryption with Associated Data (AEAD) [30]. Unique to EDHOC, however, new parameters like connection identifiers, transcript hashes, methods, and others exist. Moreover, EDHOC protocol works in two modes: asymmetric-key-based authentication technique that provides mutual authenticity via Diffie–Hellman shared ciphers and symmetric-key-based authentication that relies on preshared symmetric keys. Table 1 shows the parameters used in the EDHOC protocol.

2.3.1. Asymmetric-Key-Based EDHOC Protocol. The execution steps of an EDHOC protocol that uses asymmetric-keys are shown in Figure 2. Furthermore, to better understand and visualize the operations of both variants of the EDHOC protocol, we presented a state diagram as shown in Figures 3 and 4 . Take note that the figures show one session connection between the initiator and the responder.

(1) Initiator ⟶ Responder. Before the commencement of the protocol, the initiator I stores the domain parameters for the agreed elliptic curve, ID_CRED_I, AD_1, and AD_2. Firstly, the caller generates a method that identifies the authentication method and the associated correlation (corr) of the transport mechanism. Here, "method" and "corr" take values from 0 to 3 as described in [19]. The initiator also chooses $\mathrm{SUITES_I}$ from the list of cipher suites that an EDHOC protocol recognizes and select the connection identifier C_I. It then picks a number x to serve as an ECDH private key. Once the preliminary information is ready, it computes the ECDH public key G_X ($=G.x$) and TYPE ($=4$ * method + corr). Finally, it constructs and sends $\mathrm{Msg_1}$ containing TYPE, $\mathrm{SUITES_I}$, G_X, C_I, and AD_1 to the responder. Note that AD_1, at this time, cannot guarantee security as it is transmitted in plaintext.

(2) Responder ⟶ Initiator. Once it receives Msg1, the responder selects a cipher suite SUITES_R and the connection identifier C_R, and calculates the ECDH public key G^Y ($=G.y$) the same way the initiator calculated G^X. It then calculates the ECDH shared key G^{XY} ($=G^X.y$). Subsequently, the transcript hash TH_2 is computed by hashing the received message Msg1 and $\mathrm{data_2}$, where $\mathrm{data_2}$ consists of the session identifiers C_I, C_R, and the ECDH public G^Y. The responder uses TH_2 for authentication. It then computes an encryption key K_2 (HKDF (PRK, G^{XY})) from the pseudorandom key PRK (HKDF ("0x," G^{XY})) and the transcript hash $\mathrm{TH_2}$. Next, the responder constructs Msg2 by concatenating $\mathrm{CIPHERTEXT_2}$ and $\mathrm{data_2}$.

The former message is formed by first signing $\mathrm{CRED_R}$ and $\mathrm{TH_2}$ with the responder's private key, followed by encrypting the signature, ID_CRED_R, and AD_2 with K_2. The latter message is simply the concatenation of C_I, C_R, and G^Y. Finally, the responder sends Msg2 to the initiator.

Table 1: Symbols and notations used in the EDHOC protocol.

Components	Description
Method	One of the four types of authentication methods agreed by the initiator and the responder.
Corr	One of the four types of correlation mechanisms provided by the transport path.
$SUITES\_I$, $SUITS\_R$	List of cipher suites (in order of preference) supported by the initiator and the responder, respectively.
x, y	The ECDH ephemeral private keys of the initiator and the responder, respectively.
G_X, G_Y	The ECDH ephemeral public keys of the initiator and the responder, respectively.
p	A prime number that states the size of the finite field.
a, b	The coefficients of the elliptic curve equation.
G	The generator (base point) of the subgroup.
h, n	The cofactor and order of the subgroup, respectively.
C_I, C_R	Connection identifiers for the initiator and responder, respectively, that are used to facilitate the retrieval of the protocol state.
AD	Application data (also known as external authorization data).
$CRED_I$, $CRED_R$	The credentials containing the public authentication keys of the initiator and the responder, respectively.
ID_CRED_I, ID_CRED_R	The identifiers for the credentials $CRED_I$ and $CRED_R$, respectively.
TH	Transcript hashes used for key derivation and additional authenticated data.
K	Session key.
PRK	Pseudorandom key.
PSK	Preshared key.
AEAD (K;)	Authenticated Encryption with Associated Data using a key K.
Sig (I; .), Sig(R; .)	Digital signatures made with the private authentication key of the initiator and the responder, respectively.

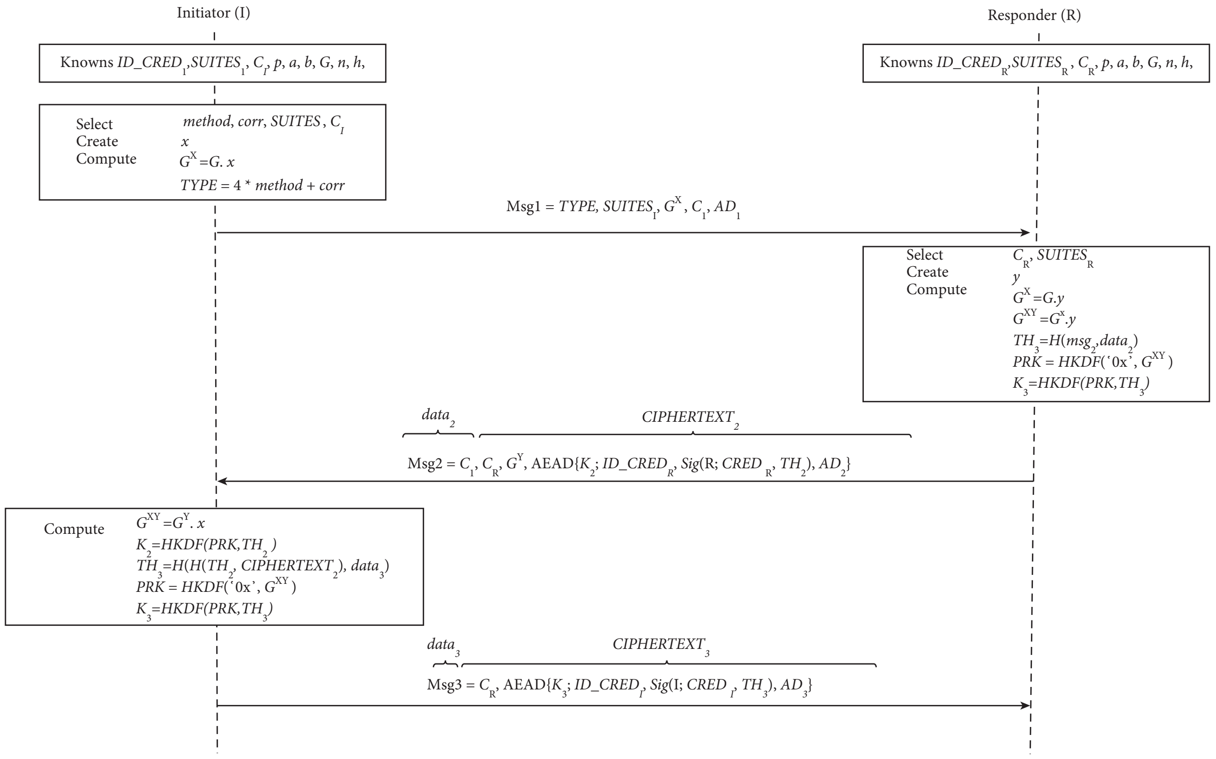

Initiator (I)
Responder (R)
Knowns ID_CRED_1,$SUITES_1$, C_I, p, a, b, G, n, h,
Knowns ID_CRED_R,$SUITES_R$, C_R, p, a, b, G, n, h,
Select
Create
Compute
$method$, $corr$, $SUITES$, C_I
x
$G^X=G.\,x$
$TYPE = 4 * method + corr$
Msg1 = $TYPE$, $SUITES_I$, G^X, C_1, AD_1
Select
Create
Compute
C_R, $SUITES_R$
y
$G^X=G.y$
$G^{XY}=G^x.y$
$TH_3=H(msg_2,data_2)$
$PRK = HKDF(\text{'0x'}, G^{XY})$
$K_3=HKDF(PRK,TH_3)$
$data_2$
$CIPHERTEXT_2$
Msg2 = C_1, C_R, G^Y, AEAD{K_2; ID_CRED_R, Sig(R; $CRED_R$, TH_2), AD_2}
Compute
$G^{XY}=G^Y.\,x$
$K_2=HKDF(PRK,TH_2)$
$TH_3=H(H(TH_2, CIPHERTEXT_2), data_3)$
$PRK = HKDF(\text{'0x'}, G^{XY})$
$K_3=HKDF(PRK,TH_3)$
$data_3$
$CIPHERTEXT_3$
Msg3 = C_R, AEAD{K_3; ID_CRED_I, Sig(I; $CRED_I$, TH_3), AD_3}

Figure 2: Asymmetric-key-based EDHOC protocol.

(3) Initiator ⟶ Responder. When Msg2 reaches the initiator, it computes the ECDH shared key G^{XY} (=$G^Y.x$), PRK, TH_2, and K_2, like the responder. Then, it uses K_2 to decrypt CIPHERTEXT_2 and retrieve ID_CRED_I, the signature, and AD_2. It then validates the signature, and if the result succeeds, it generates TH_3 and K_3. The former is constructed by first hashing CIPHERTEXT_2 with TH_2 and then rehashing the result with C_R. The latter is computed by using HKDF (PRK, TH_3). Finally, the initiator constructs a message CIPHERTEXT_3 by signing $CRED_I$ and TH_3 with its private key and encrypting it together with ID_CRED_R and AD_3 using the computed session key K_3; it forms Msg3 by concatenating C_R and CIPHERTEXT_3 and sends it to the responder. The asymmetric-key-based EDHOC protocol

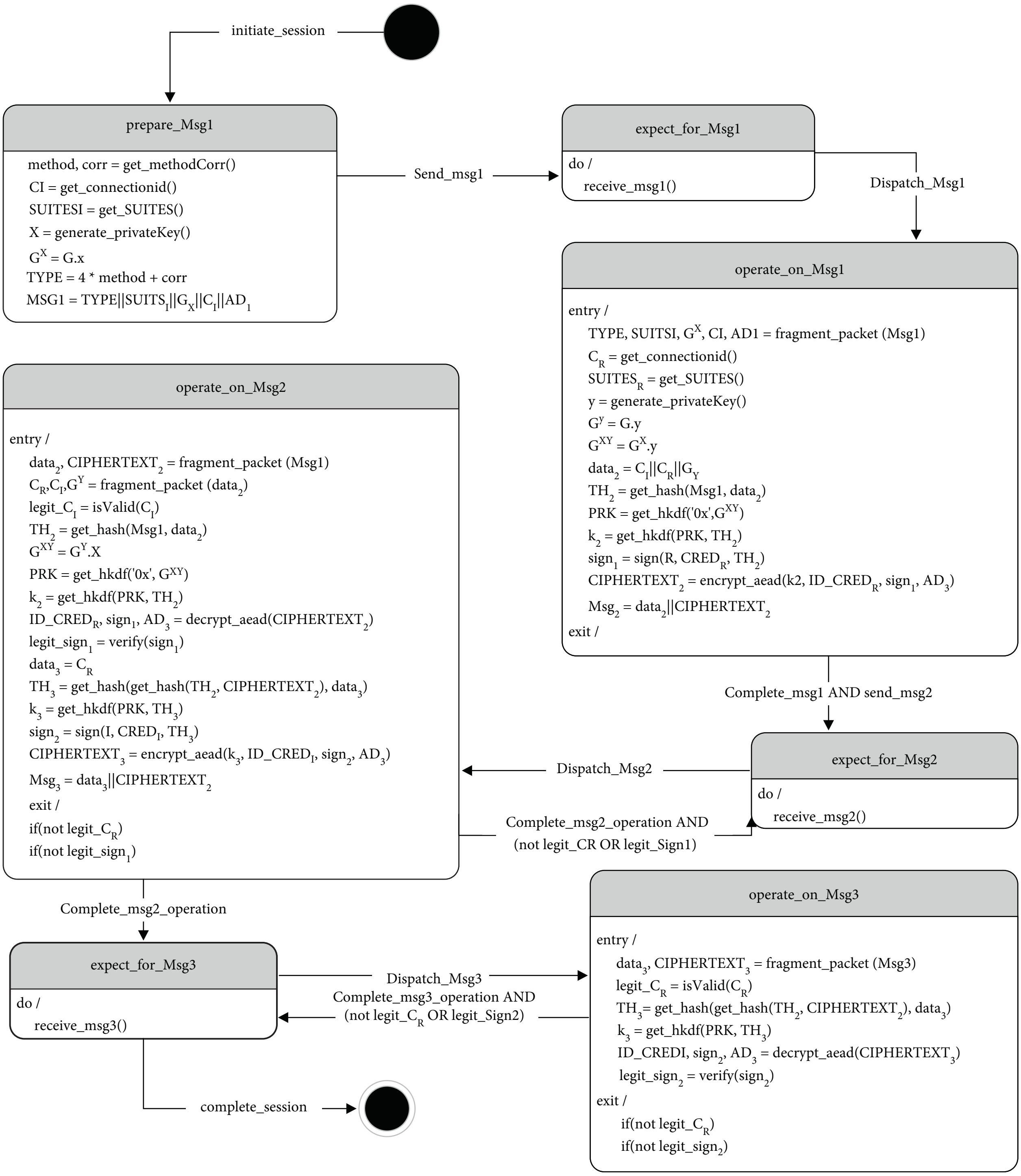

initiate_session
prepare_Msg1
method, corr = get_methodCorr()
CI = get_connectionid()
SUITESI = get_SUITES()
X = generate_privateKey()
G^X = G.x
TYPE = 4 * method + corr
MSG1 = TYPE||$SUITS_I$||G_X||C_I||AD_1
Send_msg1
expect_for_Msg1
do /
receive_msg1()
Dispatch_Msg1
operate_on_Msg1
entry /
TYPE, SUITSI, G^X, CI, AD1 = fragment_packet (Msg1)
C_R = get_connectionid()
$SUITES_R$ = get_SUITES()
y = generate_privateKey()
G^y = G.y
G^{XY} = G^X.y
$data_2$ = C_I||C_R||G_Y
TH_2 = get_hash(Msg1, $data_2$)
PRK = get_hkdf('0x',G^{XY})
k_2 = get_hkdf(PRK, TH_2)
$sign_1$ = sign(R, $CRED_R$, TH_2)
$CIPHERTEXT_2$ = encrypt_aead(k2, ID_CRED_R, $sign_1$, AD_3)
Msg_2 = $data_2$||$CIPHERTEXT_2$
exit /
Complete_msg1 AND send_msg2
expect_for_Msg2
do /
receive_msg2()
Dispatch_Msg2
operate_on_Msg2
entry /
$data_2$, $CIPHERTEXT_2$ = fragment_packet (Msg1)
C_R,C_I,G^Y = fragment_packet ($data_2$)
legit_C_I = isValid(C_I)
TH_2 = get_hash(Msg1, $data_2$)
G^{XY} = G^Y.X
PRK = get_hkdf('0x', G^{XY})
k_2 = get_hkdf(PRK, TH_2)
ID_CRED_R, $sign_1$, AD_3 = decrypt_aead($CIPHERTEXT_2$)
legit_$sign_1$ = verify($sign_1$)
$data_3$ = C_R
TH_3 = get_hash(get_hash(TH_2, $CIPHERTEXT_2$), $data_3$)
k_3 = get_hkdf(PRK, TH_3)
$sign_2$ = sign(I, $CRED_I$, TH_3)
$CIPHERTEXT_3$ = encrypt_aead(k_3, ID_CRED_I, $sign_2$, AD_3)
Msg_3 = $data_3$||$CIPHERTEXT_2$
exit /
if(not legit_C_R)
if(not legit_$sign_1$)
Complete_msg2_operation AND
(not legit_CR OR legit_Sign1)
Complete_msg2_operation
expect_for_Msg3
do /
receive_msg3()
Dispatch_Msg3
Complete_msg3_operation AND
(not legit_C_R OR legit_Sign2)
operate_on_Msg3
entry /
$data_3$, $CIPHERTEXT_3$ = fragment_packet (Msg3)
legit_C_R = isValid(C_R)
TH_3= get_hash(get_hash(TH_2, $CIPHERTEXT_2$), $data_3$)
k_3 = get_hkdf(PRK, TH_3)
ID_CREDI, $sign_2$, AD_3 = decrypt_aead($CIPHERTEXT_3$)
legit_$sign_2$ = verify($sign_2$)
exit /
if(not legit_C_R)
if(not legit_$sign_2$)
complete_session

FIGURE 3: State diagram for the asymmetric-key-based EDHOC protocol.

concludes by removing the Diffie–Hellman key pairs used to generate the encryption keys K_2 and K_3 to support perfect forward secrecy.

2.3.2. Symmetric-Key-Based EDHOC Protocol. The symmetric-key-based EDHOC protocol, shown in Figure 5, is very similar to the asymmetric-key-based protocol, with the following exceptions:

(1) The public key identifiers ID_CRED_I and ID_CRED_R are not used as part of the authenticated encryption

(2) Authentication happens via preshared key *PSK* (identified by *ID_PSK*) rather than the digital signatures used in the previous protocol

(3) The protocol session keys K_2 and K_3 are derived based on Diffie–Hellman shared keys, transcript hashes, and preshared keys *PSK*

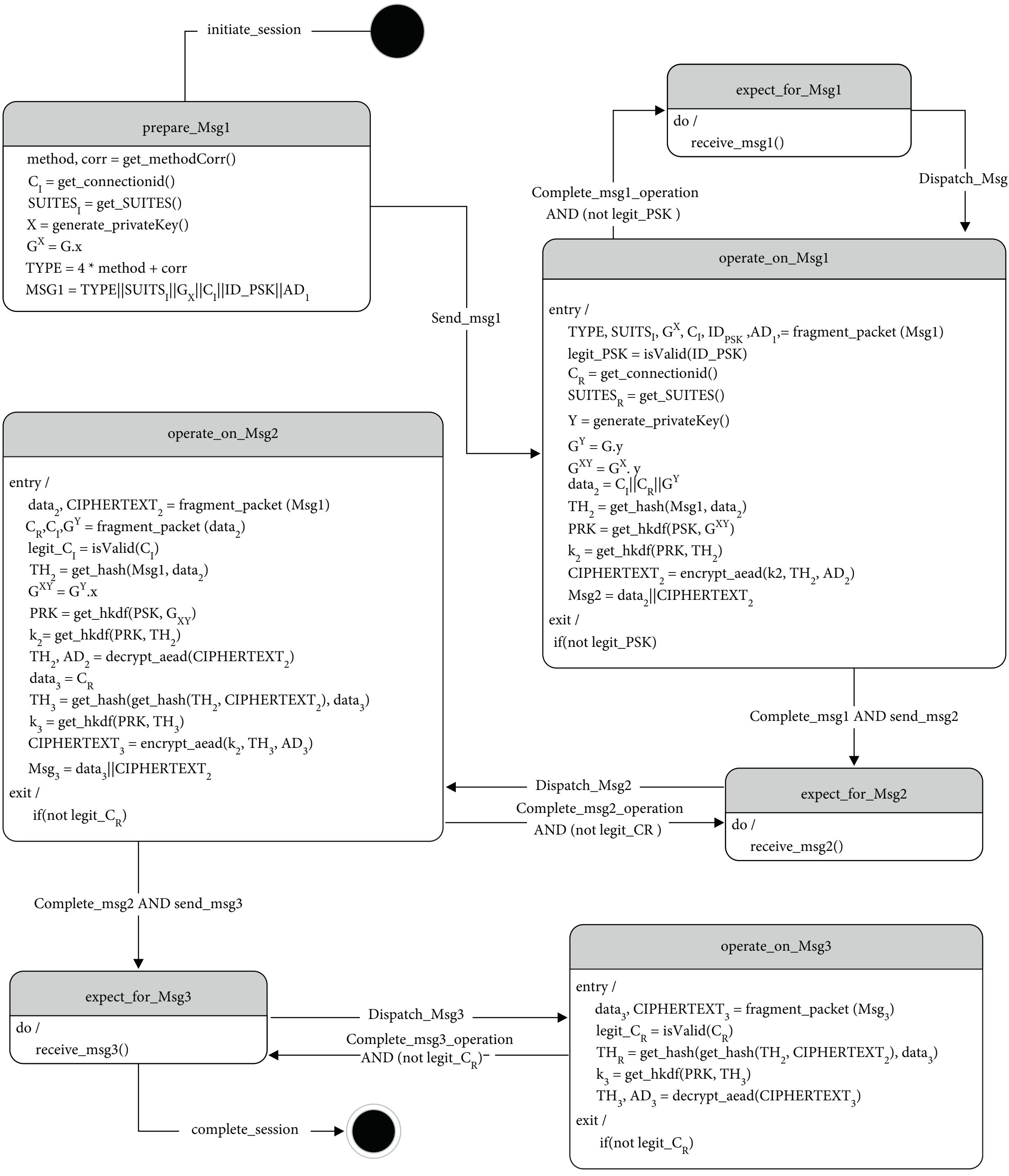

initiate_session
prepare_Msg1
method, corr = get_methodCorr()
C_I = get_connectionid()
$SUITES_I$ = get_SUITES()
X = generate_privateKey()
G^X = G.x
TYPE = 4 * method + corr
MSG1 = TYPE||$SUITS_I$||G_X||C_I||ID_PSK||AD_1
Send_msg1
expect_for_Msg1
do /
receive_msg1()
Complete_msg1_operation AND (not legit_PSK)
Dispatch_Msg1
operate_on_Msg1
entry /
TYPE, $SUITS_I$, G^X, C_I, ID_{PSK} ,AD_1,= fragment_packet (Msg1)
legit_PSK = isValid(ID_PSK)
C_R = get_connectionid()
$SUITES_R$ = get_SUITES()
Y = generate_privateKey()
G^Y = G.y
G^{XY} = G^X. y
$data_2$ = C_I||C_R||G^Y
TH_2 = get_hash(Msg1, $data_2$)
PRK = get_hkdf(PSK, G^{XY})
k_2 = get_hkdf(PRK, TH_2)
$CIPHERTEXT_2$ = encrypt_aead(k2, TH_2, AD_2)
Msg2 = $data_2$||$CIPHERTEXT_2$
exit /
if(not legit_PSK)
operate_on_Msg2
entry /
$data_2$, $CIPHERTEXT_2$ = fragment_packet (Msg1)
C_R,C_I,G^Y = fragment_packet ($data_2$)
legit_C_I = isValid(C_I)
TH_2 = get_hash(Msg1, $data_2$)
G^{XY} = G^Y.x
PRK = get_hkdf(PSK, G_{XY})
k_2= get_hkdf(PRK, TH_2)
TH_2, AD_2 = decrypt_aead($CIPHERTEXT_2$)
$data_3$ = C_R
TH_3 = get_hash(get_hash(TH_2, $CIPHERTEXT_2$), $data_3$)
k_3 = get_hkdf(PRK, TH_3)
$CIPHERTEXT_3$ = encrypt_aead(k_2, TH_3, AD_3)
Msg_3 = $data_3$||$CIPHERTEXT_2$
exit /
if(not legit_C_R)
Complete_msg1 AND send_msg2
Dispatch_Msg2
Complete_msg2_operation AND (not legit_CR)
expect_for_Msg2
do /
receive_msg2()
Complete_msg2 AND send_msg3
expect_for_Msg3
do /
receive_msg3()
Dispatch_Msg3
Complete_msg3_operation AND (not legit_C_R)
operate_on_Msg3
entry /
$data_3$, $CIPHERTEXT_3$ = fragment_packet (Msg_3)
legit_C_R = isValid(C_R)
TH_R = get_hash(get_hash(TH_2, $CIPHERTEXT_2$), $data_3$)
k_3 = get_hkdf(PRK, TH_3)
TH_3, AD_3 = decrypt_aead($CIPHERTEXT_3$)
exit /
if(not legit_C_R)
complete_session

Figure 4: State diagram for the symmetric-key-based EDHOC protocol.

3. Formal Security Verification for EDHOC Protocol

This section describes the formal security verification of both variants of the EDHOC protocol. First, we leverage BAN-Logic to analyse any security flaw that may exist in the protocol. Next, to further strengthen the verification result and complement the weakness of the first approach, we will use the AVISPA tool.

3.1. BAN-Logic-Based Formal Verification. BAN-Logic is a modal logic of beliefs (proposed by Burrows, Abadi, and Needham) used to verify authentication protocols in a formal manner [26, 31]. The formal description of the authentication process, participants' knowledge, and beliefs serve as a foundation for analysing the changes at each level of the protocol. BAN-Logic is the most utilized approach for examining various security protocols due to its simplicity and robustness.

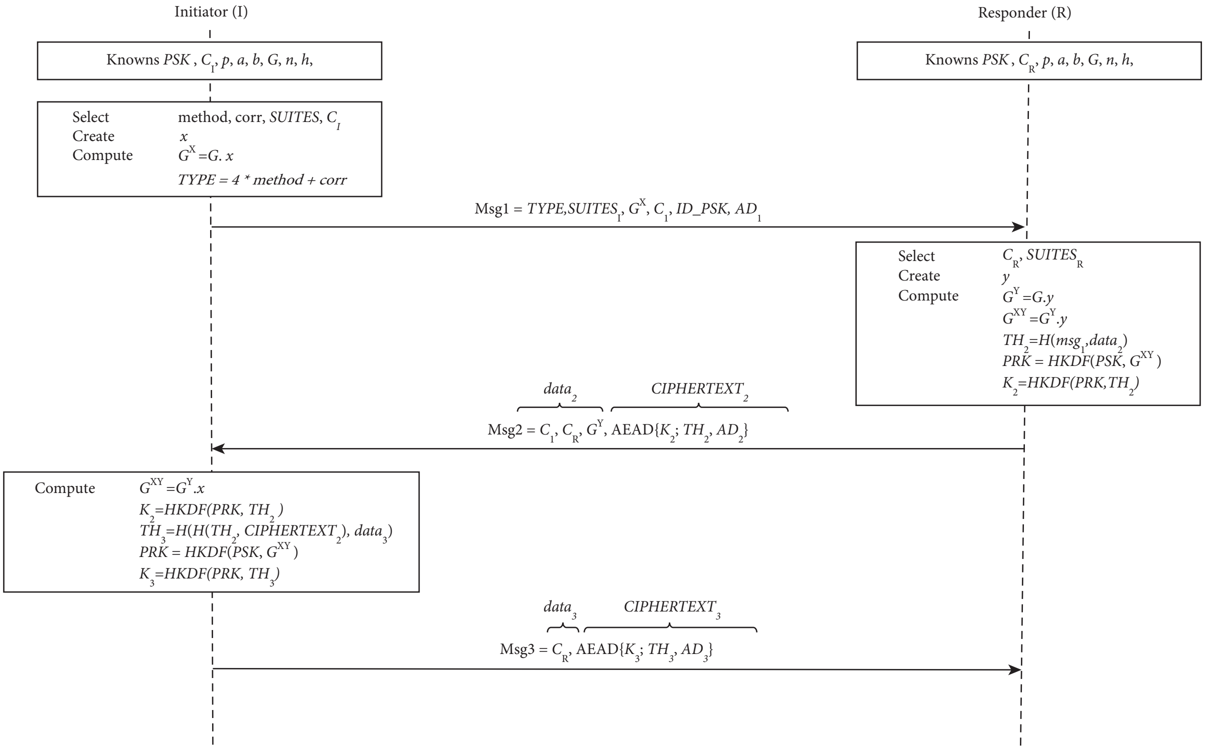


Figure 5: Symmetric-key-based EDHOC protocol.

Verification of security protocols using this method starts with converting the protocol into an idealized form through idealization. Here, only protected messages, traversing from one participant to another, are of interest. Then, realistic assumptions and security objectives that the protocol should guarantee proceed. Subsequently, the derivation of the security goals continues by applying different BAN-Logic rules, the premises, and the intermediate results of the derivation. Tables 2 and 3 describe the symbols and formulas used in the BAN-Logic formalization process, respectively.

3.1.1. The Asymmetric-Key Option

Idealization. An idealized version of the asymmetric form of the EDHOC protocol is shown below. Note that the idealized form only comprises encrypted (protected) communications, which is why Msg1 is left out.

$$R \longrightarrow I\colon \left\{ID\_CRED_R, \left\{CRED_R, \xrightarrow{G^Y} R, \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right\}_{R^{-1}}, AD_2\right\}_{G^{XY}}, \tag{1}$$

$$I \longrightarrow R\colon \left\{ID\_CRED_I, \left\{CRED_R, \xrightarrow{G^Y} R, \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right\}_{R^{-1}}, AD_3\right\}_{G^{XY}}. \tag{2}$$

Goals. The following objectives are established for verifying mutual authentication and key exchange between *I* and *R*. Consequently, while the goals in (5) and (6) show the beliefs *I* has on *R*'s trust concerning its credential identity and the associated data (respectively), (9) and (10) show the opposite. About key exchange, the goals in (3) and (4) show *I*'s belief on the session key, and (7) and (8) assert *R*'s belief on the same key.

Table 2: BAN-Logic notations.

Notation	Meaning
$R\ \textit{believes}\ M$	R believes that message M is true
R sees M	R receives message M at any point in time
R said M	R previously sent message M
R controls M	R has jurisdiction over M
$\textit{Fresh}\ (M)$	M is fresh
$R\overset{S}{\leftrightarrow} I$	S is a secret key shared between M and N
$\xrightarrow{S} R$	S is R's public key
$R\overset{S}{\Leftrightarrow}I$	S is a secret that R and I share
$\{M\}_K$	M is a message encrypted with a key K
M, V	M is combined with V

Table 3: BAN-Logic rules.

Rule name	Rule
Message meaning rule (MM)	$(R \text{ believes } R\overset{S}{\leftrightarrow} I,\ R \text{ sees } \{M\}_S\ /R \text{ believes } I \text{ said } M)$ $(R \text{ believes } R\overset{S}{\Leftrightarrow}I,\ R \text{ sees } M_S\ /R \text{ believes } I \text{ said } M)$ $(R \text{ believes } \xrightarrow{S} I,\ R \text{ sees } \{M\}_{S^{-1}}\ /R \text{ believes } I \text{ said } M)$
Nonce verification (NV) rule	$(R \text{ believes } \#\,(M), R \text{ believes } I \text{ said } M\ /R \text{ believes } I \text{ believes } M)$
Jurisdiction (JR) rule	$(R \text{ believes } I \text{ controls } K,\ R \text{ believes } I \text{ believes } K/R \text{ believes } K)$
Freshness (FR) rule	$(R \text{ believes fresh}\,(M)/R \text{ believes fresh}\,(M, Q))$
Decomposition (DR) rule	$(R \text{ sees } (M, Q)/R \text{ sees } M)$
Belief conjunction (BC) rule	$(R \text{ believes } M, R \text{ believes } Q\ /R \text{ believes } (M, Q))$ $(R \text{ believes } I \text{ believes } (M, Q)\ /R \text{ believes } I \text{ believes } M)$ $(R \text{ believes } I \text{ said } (M, Q)\ /R \text{ believes } I \text{ said } M)$
Diffie–Hellman (DH) rule	$(R \text{ believes } I \text{ said } \xrightarrow{G^M} I, R \text{ believes } \xrightarrow{G^Q} R\ /R \text{ believes } R\overset{g^{MQ}}{\leftrightarrow} I)$ $(R \text{ believes } I \text{ said } \xrightarrow{G^M} I, R \text{ believes } \xrightarrow{G^Q} R\ /R \text{ believes } R\overset{G^{MQ}}{\Leftrightarrow} G^{MQ} I)$

$$I \text{ believes } I\overset{G^{XY}}{\longleftrightarrow} R, \tag{3}$$

$$I \text{ believes } R \text{ believes } I\overset{G^{XY}}{\longleftrightarrow} R, \tag{4}$$

$$I\ \textit{believes}\ R\ \textit{believes}\ AD_2, \tag{5}$$

$$I\ \textit{believes}\ R\ \textit{believes}\ ID\_CRED_R, \tag{6}$$

$$R \text{ believes } I \overset{G^{XY}}{\leftrightarrow} R, \tag{7}$$

$$R \text{ believes } I \text{ believes } I \overset{G^{XY}}{\leftrightarrow} R, \tag{8}$$

$$R \text{ believes } I\ \textit{believes}\ AD_3 \tag{9}$$

$$R\ \textit{believes}\ I\ \textit{believes}\ ID\_CRED_I. \tag{10}$$

Assumptions. There are some assumptions and hypotheses we need to set to derive the above goals. Accordingly, the assumptions in (11), (15), and (16) show R's belief in its ECDH public key, the long-term public key of I, and the freshness of its ECDH public key. On the other hand, while (12) and (14) point out I's belief in its ECDH public key and its freshness, (13) indicates the belief "I" has in R's long-term public key. Finally, the two hypotheses (17) and (18) imply that R trusts that I sent its ECDH public key and vice versa, respectively.

$$R \text{ believes } \xrightarrow{G^Y} R, \tag{11}$$

$$I \text{ believes } \xrightarrow{G^X} I, \tag{12}$$

$$I \text{ believes } \xrightarrow{P} U(R)\ R, \tag{13}$$

$$I \text{ believes } \# \xrightarrow{G^X} I, \tag{14}$$

$$R \text{ believes } \xrightarrow{P} U(I)\ I, \tag{15}$$

$$R \text{ believes } \#\left(\xrightarrow{G^Y} R\right), \tag{16}$$

$$R \text{ believes } I \text{ said } \xrightarrow{G^X} I, \tag{17}$$

$$I \text{ believes } R \text{ said } \xrightarrow{G^Y} R. \tag{18}$$

Derivations. As a final step of the formal analysis, derivation of goals proceeds. To do so, we leverage the BAN-Logic rules (shown in Table 3), idealizations, assumptions, and the intermediate results of the derivation process. Therefore, if all goals can be derived, the target protocol is considered secure. Otherwise, the protocol may be vulnerable to threats.

$$R \text{ believes } I \overset{G^{XY}}{\leftrightarrow} R \text{ by } (11),\ (17),\ DH, \tag{19}$$

$$I \text{ believes } I \overset{G^{XY}}{\leftrightarrow} R\, by\ (12),\ (18),\ DH, \tag{20}$$

$$I \text{ sees } \left\{ID\_CRED_R,\ \left\{CRED_R,\ \xrightarrow{G^Y} R\ ,\ \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right\}_{R^{-1}},\ AD_2\right\}_{G^{XY}} \text{from} (1), \tag{21}$$

$$I \text{ believes } R \text{ said } \left[\begin{array}{c} ID_{CREDR}, \\ \left\{CRED_R,\ \xrightarrow{G^Y} R\ ,\ \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right\}_{R^{-1}},\ AD_2\right] \text{ by} (20),\ (21), MM, \end{array}\right. \tag{22}$$

$$I \text{ sees } \left\{CRED_R,\ \xrightarrow{G^Y} R\ ,\ \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right\}_{R^{-1}} \text{ by } (22), \tag{23}$$

$$I \text{ believes } R \text{ said } \left[CRED_R,\ \xrightarrow{G^Y} R\ ,\ \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right] \text{ by } (23),\ (13), MM, \tag{24}$$

$$I \text{ belives } R \text{ believes } \left[CRED_R,\ \xrightarrow{G^Y} R\ ,\ \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right] \text{ by } (24),\ (14),\ FR,\ NV, \tag{25}$$

$$I \text{ believes } R \text{ believes } I \overset{G^{XY}}{\leftrightarrow} R \text{ by } (25),\ BC, \tag{26}$$

$$I \text{ believes } R \text{ believes } \xrightarrow{G^Y} R \text{ by } (25),\ BC, \tag{27}$$

$$I \text{ believes } R \text{ believes } ID\_CRED_R \text{ by } (22),\ (14),\ FR,\ NV,\ BC, \tag{28}$$

$$I \text{ believes } R \text{ believes } AD_2 \text{ by } (22),\ (14),\ FR,\ NV,\ BC, \tag{29}$$

$$R \text{ sees } \left\{ID\_CRED_I,\ \left\{CRED_I,\ \xrightarrow{G^Y} R\ ,\ \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right\}_{R^{-1}},\ AD_3\right\}_{G^{XY}} \text{ from } (2), \tag{30}$$

$$R \text{ believes } I \text{ said } \left[ID_{CREDI},\ \left\{CRED_I,\ \xrightarrow{G^Y} R\ ,\ \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right\}_{I^{-1}},\ AD_3\right] \text{ by } (19),\ (30),\ MM, \tag{31}$$

$$R \text{ sees } \left\{CRED_I,\ \xrightarrow{G^Y} R\ ,\ \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right\}_{R^{-1}} \text{ by } (31),\ BC, \tag{32}$$

$$R \text{ believes } I \text{ said } \left[CRED_I,\ \xrightarrow{G^Y} R\ ,\ \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right] \text{ by } (32),\ (15),\ MM, \tag{33}$$

$$R \text{ believes } I \text{ believes } \left[CRED_I,\ \xrightarrow{G^Y} R\ ,\ \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right] \text{ by } (33),\ (16),\ FR,\ NV, \tag{34}$$

$$R \text{ believes } I \text{ believes } I \overset{G^{XY}}{\leftrightarrow} R \text{ by } (34),\ BC, \tag{35}$$

$$R \text{ believes } I \text{ believes } \xrightarrow{G^X} I \text{ by } (34),\ BC, \tag{36}$$

$$R \text{ believes } I \text{ believes } ID\_CRED_I \text{ by } (31),\ (16),\ FR,\ NV, \tag{37}$$

$$R \text{ believes } I \text{ believes } AD_3 \text{ by } (31),\ (16),\ FR,\ NV, \tag{38}$$

Note that, without the two hypotheses in (17) and (18), this derivation should stop before (22). In other words, only if both hypotheses are true, the proposed protocol can achieve the goals in (3)~(10). Unfortunately, they cannot hold because the two parties have no trust in each other's ECDH public key. Therefore, we conclude that asymmetric-key option is not secure.

3.1.2. The Symmetric-Key Option

Idealization. The idealization forms of the symmetric-key option of EDHOC protocol are shown as follows:

$$R \longrightarrow I: \ \{AD_2, \xrightarrow{G^Y} R, \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\}_{G^{XY}}, \tag{39}$$

$$I \longrightarrow R: \ \{AD_3, \xrightarrow{G^Y} R, \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\}_{G^{XY}}. \tag{40}$$

Goals. In general, the goals involve the guarantee of secure key exchange and mutual authentication. In the former case, while (41) and (42) form the belief of I in the ECDH session key, (44) and (45) represent the same case for R. In the latter point, the goals in (43), (46), and (47) serve to verify the mutual authentication.

$$I \text{ believes } I \overset{G^{XY}}{\leftrightarrow} R, \tag{41}$$

$$I \text{ believes } R \text{ believes } I \overset{G^{XY}}{\leftrightarrow} R, \tag{42}$$

$$I \text{ believes } R \text{ believes } AD_2, \tag{43}$$

$$R \text{ believes } I \overset{G^{XY}}{\leftrightarrow} R, \tag{44}$$

$$R \text{ believes } I \textit{ believes } I \overset{G^{XY}}{\leftrightarrow} R, \tag{45}$$

$$R \text{ believes } I \text{ believes } AD_3, \tag{46}$$

$$R \text{ believes } I \text{ believes } ID\_PSK. \tag{47}$$

Assumptions. While the assumptions in (49) and (50) show I's belief concerning its ECDH public key and its freshness (respectively), (48) and (51) do the same for R (respectively). Moreover, the symmetric-key option of the EDHOC protocol also requires the same additional hypotheses in (52) and (53) as the asymmetric-key option.

$$R \text{ believes } \xrightarrow{G^Y} R, \tag{48}$$

$$I \text{ believes } \xrightarrow{G^X} I, \tag{49}$$

$$I \text{ believes } \#\left(\xrightarrow{G^X} I\right), \tag{50}$$

$$R \text{ believes } \#\left(\xrightarrow{G^Y} R\right), \tag{51}$$

$$R \text{ believes } I \text{ said } \xrightarrow{G^X} I, \tag{52}$$

$$I \text{ believes } R \text{ said } \xrightarrow{G^X} R. \tag{53}$$

Derivations. The derivations of this variant of the EDHOC protocol proceed as follows:

$$R \text{ believes } I \overset{G^{XY}}{\leftrightarrow} R \text{ by } (48),\ (52),\ DH, \tag{54}$$

$$I \text{ believes } I \overset{G^{XY}}{\leftrightarrow} R \text{ by } (49),\ (53),\ DH, \tag{55}$$

$$I \text{ sees } \left\{AD_2, \xrightarrow{G^Y} R, \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right\}_{G^{XY}} \text{from } (39), \tag{56}$$

$$I \text{ believes } R \text{ said } \left[AD_2, \xrightarrow{G^Y} R, \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right] \text{ by } (55),\ (56),\ MM, \tag{57}$$

$$I \text{ believes } R \text{ believes } \left[AD_2, \xrightarrow{G^Y} R, \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right] \text{ by } (57),\ (50),\ FR,\ NV, \tag{58}$$

$$I \text{ believes } R \text{ believes } I \overset{G^{XY}}{\leftrightarrow} R \text{ by (58), } BC, \tag{59}$$

$$I \text{ believes } R \text{ believes } AD_2 \text{ by (58), } BC, \tag{60}$$

$$R \text{ sees } \left\{AD_3, \xrightarrow{G^Y} R, \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right\}_{G^{XY}} \text{from (40),} \tag{61}$$

$$R \text{ believes } I \text{ said } \left[AD_3, \xrightarrow{G^Y} R, \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right] \text{ by (54), (61), } MM, \tag{62}$$

$$R \text{ believes } I \text{ believes } \left[AD_3, \xrightarrow{G^Y} R, \xrightarrow{G^X} I, I \overset{G^{XY}}{\leftrightarrow} R\right] \text{ by (62), (51), } FR, NV, \tag{63}$$

$$R \text{ believes } I \text{ believes } I \overset{G^{XY}}{\leftrightarrow} R \text{ by (63), } BC, \tag{64}$$

$$R \text{ believes } I \text{ believes } \xrightarrow{G_X} I \text{ by (63), } BC, \tag{65}$$

$$R \text{ believes } I \text{ believes } AD_3 \text{ by (63), } BC, \tag{66}$$

Similar to the asymmetric-key option, the symmetric-key option can achieve the goals in (41)~(47) in the case that the two hypotheses in (52) and (53) are true. Thus, the hypotheses, which cannot be proved to be true, show that this option fails to satisfy the goals in (41)~(47). On the other hand, the last goal (47) indicating the privacy property cannot be achieved because *ID_PSK* is sent without being encrypted in the first message as shown in Figure 5.

To realize mutual authentication between "*I*" and "*R*," the former must believe the latter's ECDH public key, and it also must believe that the latter believes this key, and vice versa. That is, the derivations [(54), (55), (61), and (65)] of the asymmetric-key option and [(54), (55), (61), and (65)] of the symmetric-key option need to be satisfied. However, since all these derivations are entirely dependent on the fact that "*I*" ("*R*") believes "*R*" ("*I*") sent the ECDH public keys, it automatically follows that mutual authentication can only be fulfilled when these hypotheses are satisfied.

Messages Msg2 and Msg3, as illustrated in the idealizations in (39) and (40), use the ECDH session key to derive the AEAD encryption keys K_2 and K_3. Consequently, this can only happen through the hypotheses in (52) and (53) and is illustrated in derivations (59) and (64). Consequently, it is impossible to conclude that the session key is successfully communicated in the present asymmetric-key form of the protocol. Similarly, the symmetric version of the EDHOC protocol also fails to successfully exchange the session key without the hypotheses in (52) and (53). Thus, the derivations in (54) and (55) for "*I*" and "*R*" to believe the session key, respectively, require the use of the hypotheses.

Perfect forward secrecy is a characteristic of robust protocols because it protects previous sessions from future key compromise attempts. Accordingly, the asymmetric variant of the EDHOC protocol leverages the unique generation of ECDH private keys for each session of the protocol run to realize perfect forward secrecy. Likewise, in the symmetric-key option of the EDHOC protocol, the generation of the secret keys K_2 and K_3 uses the nonstatic Diffie–Hellman session key between "*I*" and "*R*." Thus, the symmetric-key option of the EDHOC protocol also provides perfect forward secrecy.

For both symmetric and asymmetric alternatives of EDHOC protocol to provide confidentiality and integrity, secure session key exchange must be in place. However, "*I*" and "*R*" may fail to transfer this key securely, as described earlier. As a result, the protocol cannot guarantee both confidentiality and integrity security properties.

Finally, due to the absence of authentication for the initial message, the anonymity of the responder's identifier for the public authentication keys (ID_CRED_R, for asymmetric-key option) and preshared key (*ID_PSK*, for the symmetric-key option) can be exposed.

Table 4 summarizes the result of the BAN-Logic derivation process for both options of the EDHOC protocol. As illustrated in the table and explanation above, both options of the protocol are insecure.

3.2. AVISPA-Based Formal Verification. AVISPA is an automation tool for modelling and analysing security protocols [27]. The description of the formal verification process using AVISPA proceeds as follows. First, we use a High-Level Protocol Specification Language (HLPSL) [32] to model the protocol. The HLPSL2IF component then converts the HLPSL-modelled protocols to Intermediate Format (IF). Finally, using the On-the-Fly Model-Checker (OFMC) [33], CL-based Attack Searcher (CL-AtSe) [34], SAT-based Model-Checker (SATMC) [35], and Tree-Automata-based Protocol Analyzer (TA4SP) [36], the IF is transformed to Output Format (OF). Figure 6 shows the general system

Table 4: Security property satisfaction.

No.	Security properties	Asymmetric-key option	Symmetric-key option
SP1	Mutual authentication	X	X
SP2	Secure key exchange	X	X
SP3	Perfect forward secrecy	✓	✓
SP4	Confidentiality	X	X
SP5	Integrity	X	X
SP6	Anonymity	X	X

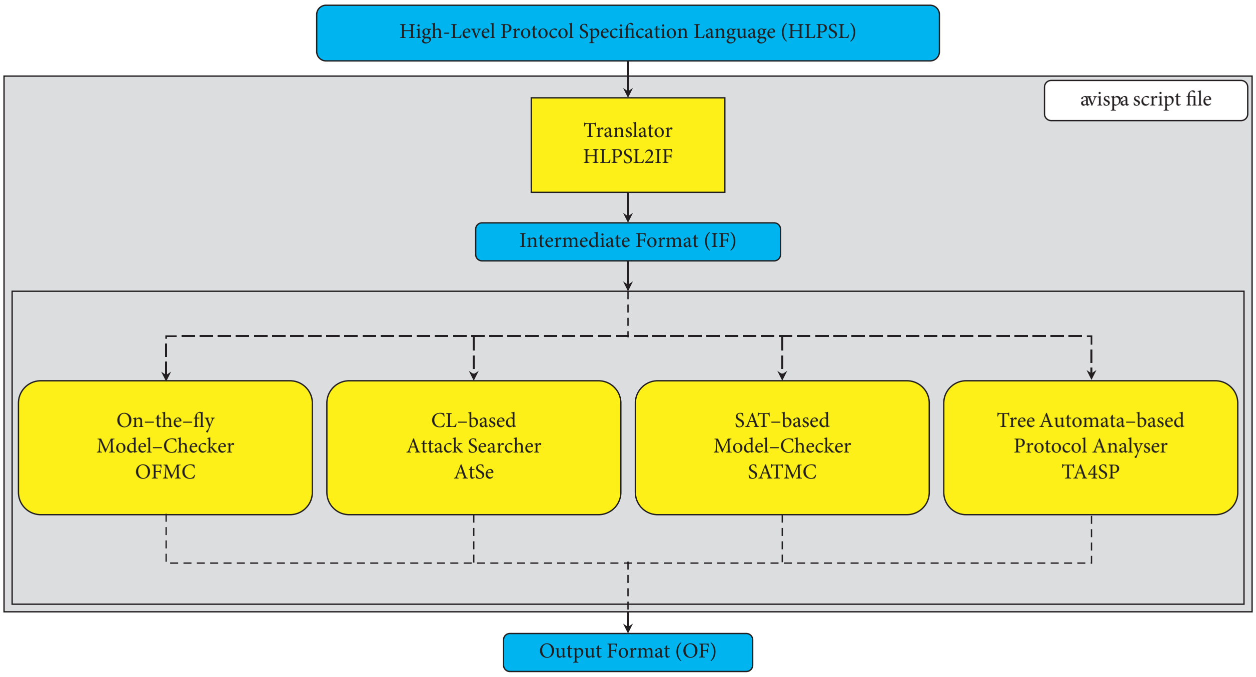


Figure 6: AVISPA system structure.

architecture of the tool, highlighting the main processes from HLPSL to OF.

HLPSL is composed of different roles such as Basic Role, Session Role, and Environment Role:

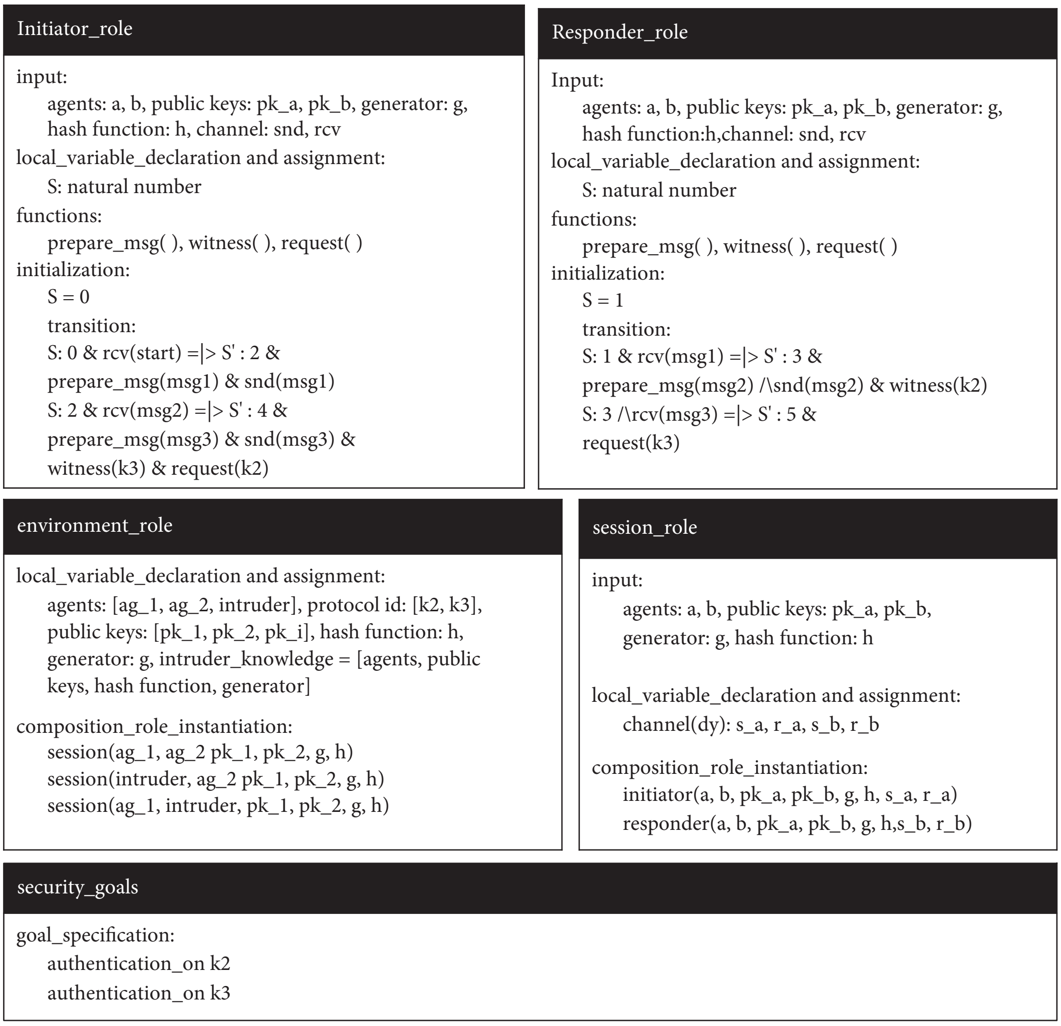
(i) *Basic Role*. This is a role that models protocol participants in a function with parameters. It consists of steps such as header expression, local variable declaration, and initialization. Additionally, it identifies communication modelling, which specifies the channel for communication between the modelled participants and indicates real-world protocol behaviour. It also defines, together with these parameters, transitions that denote message reception and the corresponding reply of the agent.

(ii) *Session Role*. This function receives the agents and other parameters to activate the previous role. It is executed via a composition to instantiate the parts in a parallel manner. /\ represents such parallel execution of prior roles.

(iii) *Environment Role*. It is a role that comprises global constants with the agents and sessions defined in the above two roles. In addition, it outlines an attacker's knowledge of the protocol's communication. The intruder's knowledge concerning the execution of the protocol is also defined. Like the Session Role, parallel execution of sessions executes with the intruder's information considered. Once the Environment Role completes, the security goals follow, and their verification proceeds with OFMC and CL-AtSe submodules.

3.2.1. *The Asymmetric-Key Option.*

At first, the asymmetric-key option of EDHOC protocol is modelled in HLPSL code. The code specifies the initiator and the responder roles with their security goals, Session Role to activate the basic roles, and finally the Environment Role. The source code for the AVISPA verification for both asymmetric and symmetric variants can be found in the Supplementary Materials (available here), while the pseudocodes are presented in Figure 7 (for asymmetric variant) and Figure 8 (for symmetric variant). The obtained verification results for the asymmetric option based on OFMC module and CL-AtSe module are also shown in Figure 9. The attack simulation of the asymmetric-key option of EDHOC protocol is illustrated in Figure 10.



```
Initiator_role
input:
    agents: a, b, public keys: pk_a, pk_b, generator: g,
    hash function: h, channel: snd, rcv
local_variable_declaration and assignment:
    S: natural number
functions:
    prepare_msg( ), witness( ), request( )
initialization:
    S = 0
    transition:
    S: 0 & rcv(start) =|> S' : 2 &
    prepare_msg(msg1) & snd(msg1)
    S: 2 & rcv(msg2) =|> S' : 4 &
    prepare_msg(msg3) & snd(msg3) &
    witness(k3) & request(k2)

Responder_role
Input:
    agents: a, b, public keys: pk_a, pk_b, generator: g,
    hash function:h,channel: snd, rcv
local_variable_declaration and assignment:
    S: natural number
functions:
    prepare_msg( ), witness( ), request( )
initialization:
    S = 1
    transition:
    S: 1 & rcv(msg1) =|> S' : 3 &
    prepare_msg(msg2) /\snd(msg2) & witness(k2)
    S: 3 /\rcv(msg3) =|> S' : 5 &
    request(k3)

environment_role
local_variable_declaration and assignment:
    agents: [ag_1, ag_2, intruder], protocol id: [k2, k3],
    public keys: [pk_1, pk_2, pk_i], hash function: h,
    generator: g, intruder_knowledge = [agents, public
    keys, hash function, generator]
composition_role_instantiation:
    session(ag_1, ag_2 pk_1, pk_2, g, h)
    session(intruder, ag_2 pk_1, pk_2, g, h)
    session(ag_1, intruder, pk_1, pk_2, g, h)

session_role
input:
    agents: a, b, public keys: pk_a, pk_b,
    generator: g, hash function: h

local_variable_declaration and assignment:
    channel(dy): s_a, r_a, s_b, r_b
composition_role_instantiation:
    initiator(a, b, pk_a, pk_b, g, h, s_a, r_a)
    responder(a, b, pk_a, pk_b, g, h,s_b, r_b)

security_goals
goal_specification:
    authentication_on k2
    authentication_on k3
```

Figure 7: A pseudocode for the AVISPA-based verification of asymmetric-key option of EDHOC protocol.

As shown in Figure 10, the attack simulation shows the asymmetric-key option of EDHOC protocol is vulnerable due to the fact that the message is sent without any verification of the sender. In other words, when the intruder sends the message of step 2, the responder should generate and calculate all the elements for communication without any proof to the user. It seems to be able to induce resource exhaustion attacks in R due to Msg2 created or modified by the attacker.

3.2.2. The Symmetric-Key Option. Like the previous case, once we translate the protocol into an HLPSL form, the AVISPA tool passes the code through the modules (such as CL-AtSe and OFMC) to check for any security flaws. Figure 11 presents the outcome of this process. According to the verification result, the symmetric-key option of the EDHOC protocol is unsafe. Figure 12 shows the possible attack simulation for the identified security flaw.

In Figure 12, when an intruder sends a message in step 2, the responder should create all the elements for communication without user authentication as Figure 10. This may deplete R's resource due to responses to numerous authentication requests from unauthorized users.

4. Results and Discussion

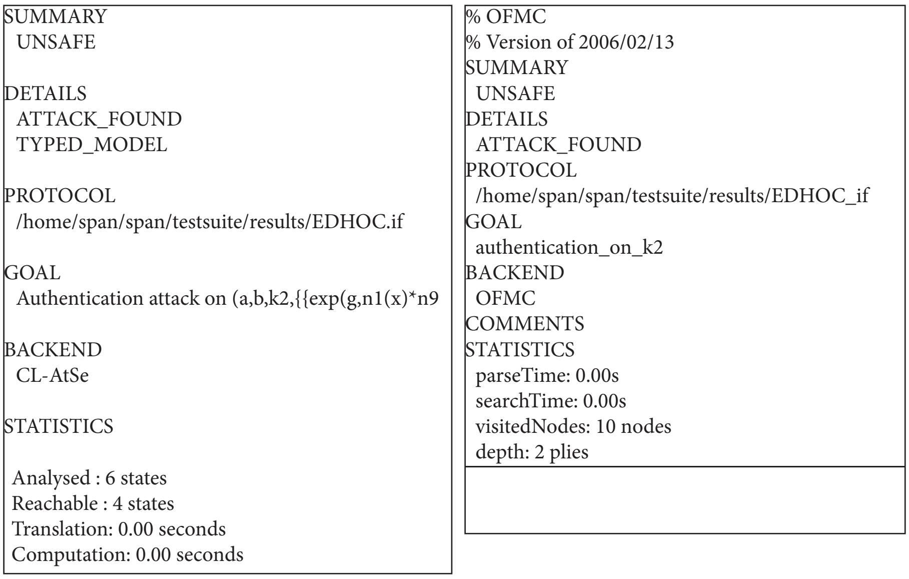
The results of the formal security analysis of both variants of the EDHOC protocol show some security-related shortcomings. In this section, we discuss these flaws.

The complete security analysis of the asymmetric-key EDHOC protocol depends on the assumption that the responder trusts the ephemeral ECDH public key G^X is from the initiator. Furthermore, the initiator also must believe that the responder sends the ephemeral ECDH public key G^Y. As shown in the BAN-Logic analysis, the hypotheses in (17) and (18) represent these two claims, respectively. Without these assumptions, mainly hypothesis (17), it is impossible to derive the goals we set. Moreover, it is worth mentioning that both hypotheses are merely there to complete the proof. Hence, it is crucial to realize them to guarantee the evidence.

Similarly, the AVISPA results for both asymmetric and symmetric variants of the protocol also show that an attack can happen (Figures 10 and 12). The responder's failure to

```
Initiator_role
Input:
    agents: a, b, symmetric key: psk, generator: g,
    hash function: h, channel: snd, rcv
local_variable_declaration and assignment:
    S: natural number
functions:
    prepare_msg( ), witness( ), request( )
initialization:
    S = 0
    transition:
    S: 0 & rcv(start) =|> S' : 2 &
    prepare_msg(msg1) & snd(msg1)
    S: 2 & rcv(msg2) =|> S' : 4 &
    prepare_msg(msg3) & snd(msg3) &witness(k3) & request(k2)

Responder_role
Input:
    agents: a, b, symmetric key: psk, generator: g,
    hash function: h, channel: snd, rcv
local_variable_declaration and assignment:
    S: natural number
functions:
    prepare_msg( ), witness( ), request( )
initialization:
    S = 1
    transition:
    S: 1 & rcv(msg1) =|> S' : 3 &
    prepare_msg(msg2) & snd(msg2) & witness(k2)
    S: 3 & rcv(msg3) =|> S' : 5 & request(k3)

environment_role
local_variable_declaration and assignment:
    agents: [ag_1, ag_2, intruder], symmetric key: psk,
    hash function: h, generator: g, protocol id: [k2, k3],
    intruder_knowledge = [agents, symmetric key,
    hash function, generator]
composition_role_instantiation:
    session(ag_1, ag_2, psk, g, h)
    session(intruder, ag_2, psk, g, h)
    session(ag_1, intruder, psk, g, h)

session_role
input:
    agents: a, b, symmetric key: psk, generator: g,
    hash function: h
local_variable_declaration and assignment:
    channel(dy): s_a, r_a, s_b, r_b
composition_role_instantiation:
    initiator(a, b, pk_a, pk_b, g, h, s_a, r_a)
    responder(a, b, pk_a, pk_b, g, h, s_b, r_b)

security_goals
goal_specification:
    authentication_on k2
    authentication_on k3
```

Figure 8: A pseudocode for the AVISPA-based verification of symmetric-key option of EDHOC protocol.



```
SUMMARY
  UNSAFE

DETAILS
  ATTACK_FOUND
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/EDHOC.if

GOAL
  Authentication attack on (a,b,k2,{{exp(g,n1(x)*n9

BACKEND
  CL-AtSe

STATISTICS

 Analysed : 6 states
 Reachable : 4 states
 Translation: 0.00 seconds
 Computation: 0.00 seconds
```

```
% OFMC
% Version of 2006/02/13
SUMMARY
 UNSAFE
DETAILS
 ATTACK_FOUND
PROTOCOL
 /home/span/span/testsuite/results/EDHOC_if
GOAL
 authentication_on_k2
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 0.00s
 visitedNodes: 10 nodes
 depth: 2 plies
```

Figure 9: Verification result of the asymmetric-key option of EDHOC protocol.

ensure the integrity of Msg1 and the difficulty of the initiator in validating Msg2 are the significant reasons for this attack. Especially for the latter point, given that the generation of the secret key K_2 depends on the ECDH session key G^{XY}, the initiator has no option but to trust the responder's ECDH public key G^Y transmitted in plaintext to verify Msg2.

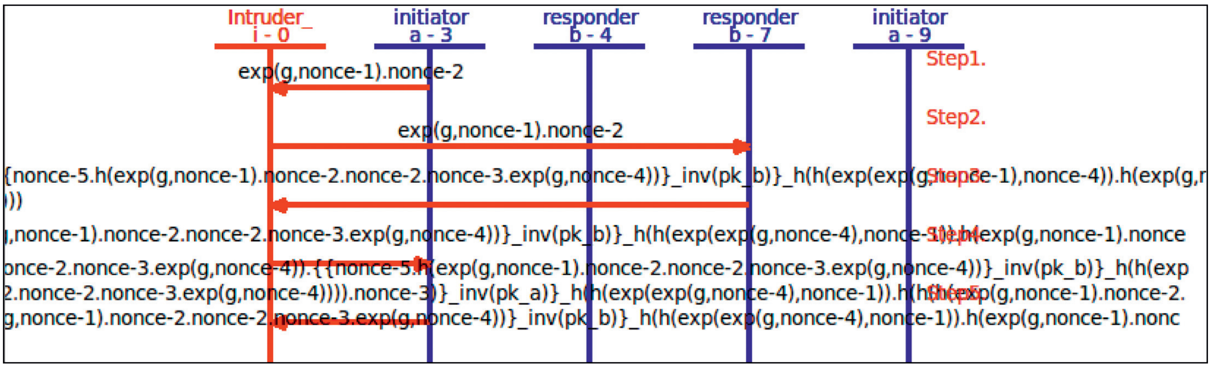
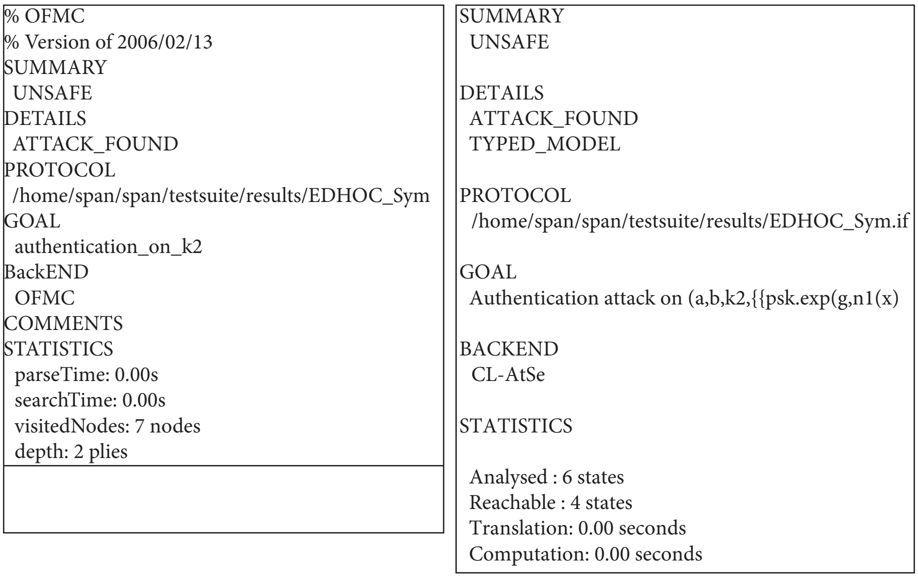

Intruder_
i - 0
initiator
a - 3
responder
b - 4
responder
b - 7
initiator
a - 9
Step1.
exp(g,nonce-1).nonce-2
Step2.
exp(g,nonce-1).nonce-2
Step3.
{nonce-5.h(exp(g,nonce-1).nonce-2.nonce-2.nonce-3.exp(g,nonce-4))}_inv(pk_b)}_h(h(exp(exp(g,nonce-1),nonce-4)).h(exp(g,r
)))
Step4.
,nonce-1).nonce-2.nonce-2.nonce-3.exp(g,nonce-4))}_inv(pk_b)}_h(h(exp(exp(g,nonce-4),nonce-1)).h(exp(g,nonce-1).nonce
Step5.
once-2.nonce-3.exp(g,nonce-4)).{{nonce-5.h(exp(g,nonce-1).nonce-2.nonce-2.nonce-3.exp(g,nonce-4))}_inv(pk_b)}_h(h(exp
2.nonce-2.nonce-3.exp(g,nonce-4)))).nonce-3)}_inv(pk_a)}_h(h(exp(exp(g,nonce-4),nonce-1)).h(h(exp(g,nonce-1).nonce-2.
g,nonce-1).nonce-2.nonce-2.nonce-3.exp(g,nonce-4))}_inv(pk_b)}_h(h(exp(exp(g,nonce-4),nonce-1)).h(exp(g,nonce-1).nonc

Figure 10: Attack simulation of the asymmetric-key option of EDHOC protocol.



```
% OFMC
% Version of 2006/02/13
SUMMARY
 UNSAFE
DETAILS
 ATTACK_FOUND
PROTOCOL
 /home/span/span/testsuite/results/EDHOC_Sym
GOAL
 authentication_on_k2
BackEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 0.00s
 visitedNodes: 7 nodes
 depth: 2 plies
```

```
SUMMARY
 UNSAFE

DETAILS
 ATTACK_FOUND
 TYPED_MODEL

PROTOCOL
 /home/span/span/testsuite/results/EDHOC_Sym.if

GOAL
 Authentication attack on (a,b,k2,{{psk.exp(g,n1(x)

BACKEND
 CL-AtSe

STATISTICS

 Analysed : 6 states
 Reachable : 4 states
 Translation: 0.00 seconds
 Computation: 0.00 seconds
```

Figure 11: Verification result of symmetric-key option of EDHOC protocol.

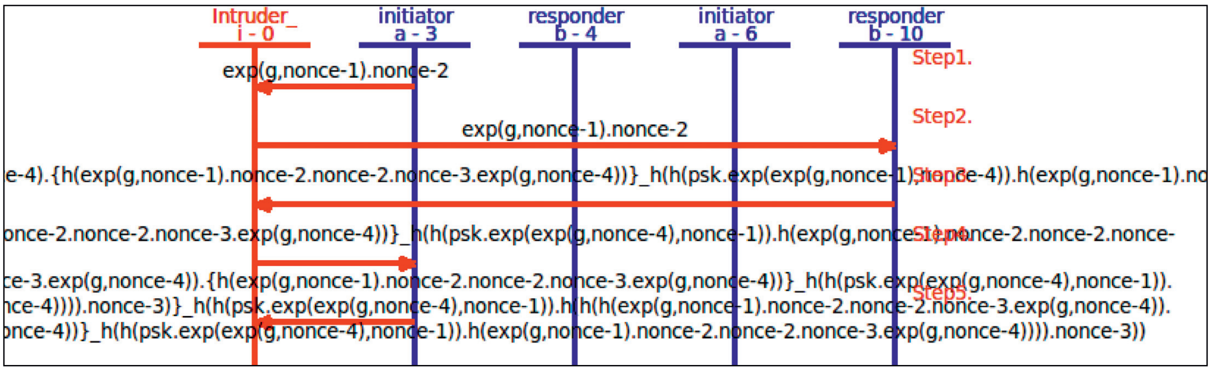

Intruder_
i - 0
initiator
a - 3
responder
b - 4
initiator
a - 6
responder
b - 10
Step1.
exp(g,nonce-1).nonce-2
Step2.
exp(g,nonce-1).nonce-2
Step3.
e-4).{h(exp(g,nonce-1).nonce-2.nonce-2.nonce-3.exp(g,nonce-4))}_h(h(psk.exp(exp(g,nonce-1),nonce-4)).h(exp(g,nonce-1).no
Step4.
once-2.nonce-2.nonce-3.exp(g,nonce-4))}_h(h(psk.exp(exp(g,nonce-4),nonce-1)).h(exp(g,nonce-1).nonce-2.nonce-2.nonce-
Step5.
ce-3.exp(g,nonce-4)).{h(exp(g,nonce-1).nonce-2.nonce-2.nonce-3.exp(g,nonce-4))}_h(h(psk.exp(exp(g,nonce-4),nonce-1)).
nce-4)))).nonce-3)}_h(h(psk.exp(exp(g,nonce-4),nonce-1)).h(h(h(exp(g,nonce-1).nonce-2.nonce-2.nonce-3.exp(g,nonce-4)).
once-4))}_h(h(psk.exp(exp(g,nonce-4),nonce-1)).h(exp(g,nonce-1).nonce-2.nonce-2.nonce-3.exp(g,nonce-4)))).nonce-3))

Figure 12: Attack simulation of the symmetric-key option of EDHOC protocol.

Another critical security threat refers to the denial-of-service attacks (more specifically, the resource exhaustion attack). Given the IoT devices' severe resource limitations concerning computation, storage, and communication, an attacker can send a significant amount of Msg1 to the responder. The responder then performs expensive operations such as encryption, signature, and key derivation functions for each of these messages before authenticating the initiator. Consequently, the responder can get easily overwhelmed by the traffic, deplete its energy, and finally cease communicating with the other end.

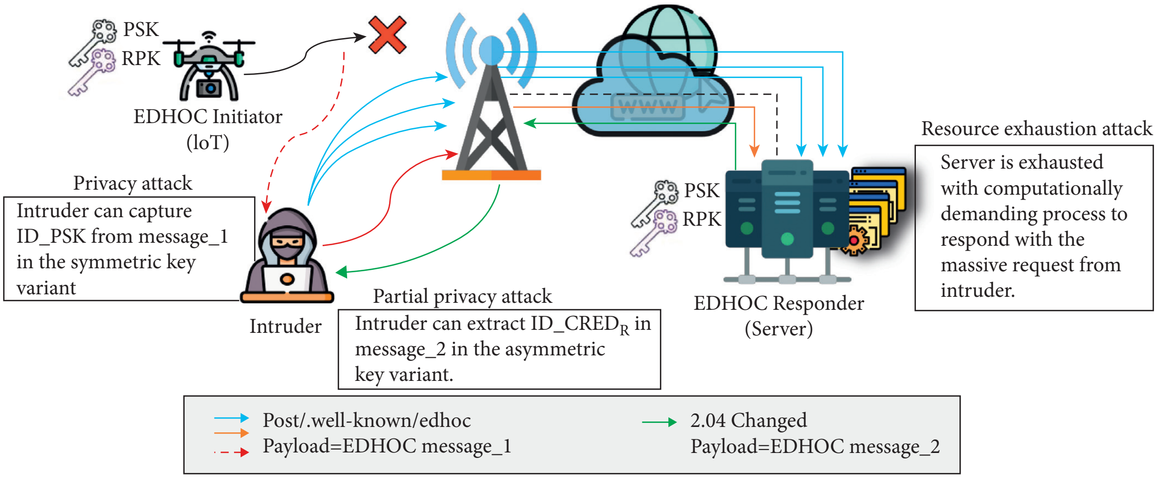

PSK
RPK
EDHOC Initiator (loT)
Privacy attack
Intruder can capture ID_PSK from message_1 in the symmetric key variant
Intruder
Partial privacy attack
Intruder can extract ID_CRED$_R$ in message_2 in the asymmetric key variant.
PSK
RPK
EDHOC Responder (Server)
Resource exhaustion attack
Server is exhausted with computationally demanding process to respond with the massive request from intruder.
Post/.well-known/edhoc
Payload=EDHOC message_1
2.04 Changed
Payload=EDHOC message_2

Figure 13: Attack simulation of the symmetric-key option of EDHOC protocol.

Table 5: Summary of related works.

Papers	Identified security issues	EDHOC version	Analysis tools used
[22]	(i) Disclosure of the responders identity in the asymmetric variant of the EDHOC protocol. (ii) An attacker can associate numerous sessions and perform attacks for the symmetric variant of the EDHOC protocol by using the same preshared key identifier. (iii) Only AD3 (for both symmetric and asymmetric variants) satisfies secrecy, perfect forward secrecy, and integrity at both the time of message arrival and the conclusion of the protocol.	Draft-selander-ace-cose-ecdhe-08 [20]	ProVerif [24]
[23]	(i) Absence of nonrepudiation security property. (ii) Lack of verification of ID_CRED$_R$ of Msg2 by the initiator. (iii) When the responder rejects recommended cipher suites, a security concern might arise because of a lengthy metasession spanning many EDHOC sessions.	Draft-selander-lake-edhoc-01 [21]	Tamarin [25]
Ours	(i) A resource exhaustion attack due to a significant amount of Msg1 sent to the responder. The responder does not authenticate Msg1 before computing expensive operations, hence depleting its resources. (ii) The responder's failure to ensure the integrity of Msg1 and the difficulty of the initiator in validating Msg2 threaten the security of the protocol. (iii) A partial privacy attack that exposes the responder's identity. Beside the mere violation of the secrecy of the responder's distinctiveness, it can enable the attacker to reduce the difficulty of stealing the public authentication keys by one step. Moreover, the privacy of ID_PSK, in symmetric-key option, is also violated as it is transmitted in plain text.	Draft-ietf-lake-edhoc-07 [19]	BAN-Logic and AVISPA [26, 27]

A second serious threat with the asymmetric-key-based EDHOC protocol, which we referred to as a partial privacy attack, is related to the privacy of *ID_CRED$_R$*. The access of the credentials containing the public authentication keys of both initiator and responder is via their identities. These identities (*ID_CRED$_I$* and *ID_CRED$_R$*), although they do not have any cryptographic purpose in the protocol, serve an essential purpose by facilitating the retrieval of the public authentication keys. Moreover, according to the standard, their privacy is protected by the session key computed by the initiator and responder. Thus, an attacker can easily break the privacy of *ID_CRED$_R$* as it can establish the session key K_2 with the responder. Therefore, it implies a privacy disclosure of one of the two identities, hence a partial privacy attack. Concerning the symmetric variant, a clear violation of privacy also happens as *ID_PSK* in Msg1 is transmitted in plain text.

It is important to note that if an attacker exploits these vulnerabilities, the results might be disastrous. For example, a medical IoT device attempting to obtain a remote service, perhaps for remote diagnostics, may fail owing to a resource depletion assault on the other end. Moreover, in cases where the responder is a sensitive medical IoT device, its identity (the identity of the credentials containing the public authentication keys) can be traced by an attacker that he/she may use to track and localize the patient eventually. Both resource exhaustion and privacy attacks (as part of transporting EDHOC via CoAP message exchanges) are shown in Figure 13. In addition, Table 5 summarizes the security issues identified by the related works together with the ones we identified.

It is critical to fix the highlighted security vulnerabilities before using EDHOC as a lightweight authenticated key exchange mechanism. Privacy-related threats are mainly initiated because the first message, from the initiator to the responder, is not authenticated. Hence, a preliminary authentication mechanism must be implemented. The responder and initiator can additionally guarantee the validity of Msg1 and Msg2 by using public-key certificates. In the case of a protracted metasession spanning several EDHOC sessions due to cipher suite rejection, the responder shall provide a mechanism that prevents the same initiator from resubmitting a new cipher suite proposal in the same session more than twice. Leveraging HMAC and timestamps can serve a good purpose in thwarting resource exhaustion attacks, as they let the responder first check the validity of the received message before performing computationally demanding instructions.

5. Conclusions

Although the rapid growth of the Internet of Things (IoT) technology is bringing a significant impact on society, efficient security protocols that are aware of the unique characteristics of IoT devices are still in their infant stage. With this regard, IETF is in progress to standardize one application layer protocol (known as EDHOC) that can assist secure communication across IoT devices while remaining lightweight. Consequently, in this paper, we formally analysed the security of this protocol using BAN-Logic and AVISPA to investigate its resilience to withstand attacks. The results show that both variants of the protocol have some serious security and privacy flaws. Primarily, a resource exhaustion attack that violates the availability of a responder's service by depleting its resources over expensive cryptographic operations such as encryption and signature can result. Next, an attacker can easily break the privacy of ID_CRED_R as it can establish the session key K_2 with the responder, which results in partial privacy disclosure of the responder's identity. A similar attack can happen when an attacker captures *ID_PSK* in the symmetric-key option of the protocol. Furthermore, an attacker can use the responder's failure to verify the integrity of Msg1 and the difficulties of the initiator in validating Msg2. Finally, we recommend that the protocol should consider authenticating the first message and provide a way to validate the second message while offering a solution to protect against resource exhaustion attacks. In future works, the authors would like to develop efficient solutions to mitigate these attacks while maintaining the lightweight nature of the EDHOC protocol.

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.

Acknowledgments

This work was partly supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (no. 2018-0-00231, Development of Context Adaptive Security Autonomous Enforcement Technology to Prevent Spread of the IoT Infrastructure Attacks, 95%) and the Soonchunhyang University Research Fund (5%).

Supplementary Materials

The AVISPA validation codes written in HLPSL for asymmetric and symmetric options are provided as separate files. (*Supplementary Materials*)

References

[1] V. Korzhuk, A. Groznykh, A. Menshikov, and M. Strecker, "Identification of attacks against wireless sensor networks based on behavior analysis," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 10, no. 2, pp. 1–21, 2019.

[2] Z.-K. Zhang, M. C. Yi Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, and S. Shieh, "IoT security: ongoing challenges and research opportunities," in *Proceedings of the 2014 IEEE 7th International Conference on IEEE*, pp. 230–234, Matsue, Japan, November 2014.

[3] Machina Research, *Press Release: Global Internet of Things Market To Grow to 27 Billion Devices, Generating USD3 Trillion Revenue in 2025*Machina Research, Stamford, CT, USA, 2021, https://bit.ly/3aHu1QG.

[4] IDC Corporate USA, *IoT Growth Demands Rethink of Long-Term Storage Strategies, Says IDC*, IDC Corporate USA, Needham, MA, USA, 2021, https://www.idc.com/getdoc.jsp?containerId=prAP46737220.

[5] J. Kim, J. Lee, J. Kim, and J. Yun, "M2M service platforms: survey, issues, and enabling technologies," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 61–76, 2014.

[6] M. Alizadeh, K. Andersson, and O. Schelén, "A survey of secure Internet of things in relation to blockchain," *Journal of Internet Services and Information Security*, vol. 10, no. 3, pp. 47–75, 2020.

[7] Y. M. Khamayseh, W. Mardini, M. Aldwairi, and H. T. Mouftah, "On the optimality of route selection in grid wireless sensor networks: theory and applications," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 11, no. 2, pp. 87–105, 2020.

[8] B. Sim and D. Han, "A study on the side-channel analysis trends for application to IoT devices," *Journal of Internet Services and Information Security*, vol. 10, no. 1, pp. 2–21, 2020.

[9] Z. Shelby, K. Hartke, and C. Bormann, *The Constrained Application Protocol (CoAP)*, IETF RFC 7252, Fremont, CA, USA, 2014, https://datatracker.ietf.org/doc/html/rfc7252.

[10] C. Bormann and P. Hoffman, *Concise Binary Object Representation (CBOR)*, IETF RFC 8949, Fremont, CA, USA, 2020, https://datatracker.ietf.org/doc/html/rfc8949.

[11] A. Minaburo, L. Toutain, C. Gomez, D. Barthel, and JC. Zuniga, *SCHC: Generic Framework for Static Context Header Compression and Fragmentation*, IETF RFC 8724, Fremont, CA, USA, 2020, https://datatracker.ietf.org/doc/html/rfc8724.

[12] T. Alladi, V. Chamola, B. Sikdar, and K.-K. R. Choo, "Consumer IoT: security vulnerability case studies and solutions," *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 17–25, 2020.

[13] J. Sanchez-Gomez, D. Garcia-Carrillo, R. Sanchez-Iborra et al., "Integrating LPWAN technologies in the 5G ecosystem: a survey on security challenges and solutions," *IEEE Access*, vol. 8, 2020.

[14] J. Sanchez-Gomez, J. Gallego-Madrid, R. Sanchez-Iborra, J. Santa, and A. Skarmeta, "Impact of SCHC compression and fragmentation in LPWAN: a case study with LoRaWAN," *Sensors*, vol. 20, no. 1, p. 280, 2020.

[15] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen, *Internet Key Exchange Protocol Version 2 (IKEv2)*, IETF RFC 7296, Fremont, CA, USA, 2014, https://datatracker.ietf.org/doc/html/rfc7296.

[16] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, IETF RFC 8446, Fremont, CA, USA, 2018, https://datatracker.ietf.org/doc/html/rfc8446.

[17] E. Rescorla, H. Tschofenig, and N. Modadugu, *The Datagram Transport Layer Security (DTLS) Protocol Version 1.3*, IETF Internet-Draft draft-ietf-tls-dtls13-43, Fremont, CA, USA, 2021, https://datatracker.ietf.org/doc/draft-ietf-tls-dtls13/.

[18] G. Selander, J. Mattsson, F. Palombini, and L. Seitz, *Object Security for Constrained RESTful Environments (OSCORE)*, IETF RFC 8613, Fremont, CA, USA, 2019, https://www.rfc-editor.org/rfc/rfc8613.html.

[19] G. Selander, J. Mattsson, and F. Palombini, *Ephemeral Diffie-Hellman over COSE (EDHOC)*, IETF Internet-Draft Draft-Ietf-Lake-Edhoc-07, Fremont, CA, USA, 2021, https://datatracker.ietf.org/doc/html/draft-ietf-lake-edhoc-07.

[20] G. Selander, J. Mattsson, F. Palombini, and L. Seitz, *Ephemeral Diffie-Hellman over COSE (EDHOC)*, IETF Internet-Draft Draft-Selander-ace-cose-ecdhe-08, Fremont, CA, USA, 2018, https://tools.ietf.org/pdf/draft-selander-ace-cose-ecdhe-08.pdf.

[21] G. Selander, J. Mattsson, F. Palombini, and L. Seitz, *Ephemeral Diffie-Hellman over COSE (EDHOC)*, IETF Internet-Draft Draft-Selander-Lake-Edhoc-01, Fremont, CA, USA, 2020, https://tools.ietf.org/pdf/draft-selander-lake-edhoc-01.pdf.

[22] A. Bruni, T. S. Jørgensen, T. G. Petersen, and C. Schürmann, "Formal verification of ephemeral Diffie-Hellman over COSE (EDHOC)," in *Proceedings of the 4th International Conference on Research in Security Standardisation*, pp. 21–36, Darmstadt, Germany, November 2018.

[23] K. Norrman, V. Sundararajan, and A. Bruni, "Formal analysis of EDHOC key establishment for constrained IoT devices," 2020, https://arxiv.org/abs/2007.11427.

[24] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, "Proverif 2.00: automatic cryptographic protocol verifier, user manual and tutorial," 2018, https://bensmyth.com/publications/2010-ProVerif-manual-version-2.00.

[25] S. Meier, B. Schmidt, C. Cremers, and D. Basin, "The TAMARIN prover for the symbolic analysis of security protocols," *Computer Aided Verification*, vol. 8044, pp. 696–701, 2013.

[26] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[27] A. Armando, D. Basin, Y. Boichut et al., "The AVISPA tool for the automated validation of Internet security protocols and applications," *Computer Aided Verification*, vol. 3576, pp. 281–285, 2005.

[28] J. Schaad, *"CBOR Object Signing and Encryption (COSE)*, IETF RFC 8152, Fremont, CA, USA, 2017, https://datatracker.ietf.org/doc/html/rfc8152 accessed on.

[29] H. Krawczyk, "SIGMA: the "SIGn-and-MAc" approach to authenticated Diffie-Hellman and its use in the IKE protocols," in *Proceedings of the 23rd Annual International Cryptology Conference (CRYPTO'03)*, vol. 2729, pp. 400–425, Santa Barbara, CA, USA, August 2003.

[30] P. Rogaway, "Authenticated-encryption with associated-data," in *Proceedings of the 9th ACM Conference on Computer and Communication Security (CCS'02)*, pp. 98–107, ACM, Washington, DC, USA, November 2002.

[31] P. Syverson and I. Cervesato, "The logic of authentication protocols," in *Foundations of Security Analysis and Design*, vol. 2171, pp. 63–137, Springer, Berlin, Germany, 2001.

[32] Y. Chevalier, L. Compagna, J. Cuellar et al., "A high level protocol specification language for industrial security-sensitive protocols," in *Proceedings of the 2004 Workshop on Specification and Automated Processing of Security Requirements (SAPS'04)*, Austrian Computer Society, Linz, Austria, September 2004.

[33] D. Basin, S. Mödersheim, and L. Viganò, "OFMC: a symbolic model checker for security protocols," *International Journal of Information Security*, vol. 4, no. 3, pp. 181–208, 2005.

[34] M. Turuani, "The CL-atse protocol analyser," in *Lecture Notes in Computer Science*, vol. 4098, pp. 277–286, Springer, Berlin, Germany, 2006.

[35] A. Armando and L. Compagna, "SATMC: a SAT-based model checker for security protocols," in *Logics in Artificial Intelligence*, vol. 3229, pp. 730–733, Springer, Berlin, Germany, 2004.

[36] Y. Boichut, P.-C. Héam, O. Kouchnarenko, and F. Oehl, "Improvements on the genet and Klay technique to automatically verify security protocols," in *Proceedings of the 3rd International Workshop on Automated Verification of Infinite State Systems (AVIS'04)*, pp. 1–11, Barcelona, Spain, April 2004.