lwEPSep: A Lightweight End-to-end Privacy-preserving Security Protocol for CTI Sharing in IoT Environments

Hoonyong Park, Jiyoon Kim, Sangmin Lee, Daniel Gerbi Duguma, Ilsun You

Department of Information Security Engineering, Soonchunhyang University, Republic of Korea
{hoon4569, 74jykim, kco0721, danielgerbi2005, ilsunu}@gmail.com

Abstract

The Internet of Things (IoT) is vulnerable to a wide range of security risks, which can be effectively mitigated by applying Cyber Threat Intelligence (CTI) sharing as a proactive mitigation approach. In realizing CTI sharing, it is of paramount importance to guarantee end-to-end protection of the shared information as unauthorized disclosure of CTI is disastrous for organizations using IoT. Furthermore, resource-constrained devices should be supported through lightweight operations. Unfortunately, the aforementioned are not satisfied by the Hypertext Transfer Protocol Secure (HTTPS), which state-of-the-art CTI sharing systems mainly depends on. As a promising alternative to HTTPS, Ephemeral Diffie-Hellman over COSE (EDHOC) can be considered because it meets the above requirements. However, EDHOC in its current version contains several security flaws, most notably due to the unprotected initial message. Consequently, we propose a lightweight end-to-end privacy-preserving security protocol that improves the existing draft EDHOC protocol by utilizing previously shared keys and keying materials while providing ticket-based optimized re-authentication. The proposed protocol is not only formally validated through BAN-logic and AVISPA, but also proved to fulfill essential security properties such as mutual authentication, secure key exchange, perfect forward secrecy, anonymity, confidentiality, and integrity. Also, comparing the protocol's performance to that of the EDHOC protocol reveals a substantial improvement with a single roundtrip to allow frequent CTI sharing.

Keywords: CTI, TAXII, EDHOC, End-to-End security, Formal verification

1 Introduction

Today's cybersecurity and threats resemble real-world warfare as network structures, attack methods, and functions diversify [1]. However, past and present experience shows that responding to every cyberattack is very inefficient and wasteful. Also, the skilled human resources and budget to make this war are not enough for everyone [2]. Therefore, CTI (Cyber Threat Intelligence) is designed to collect and share information about cyber threats and attackers to mitigate cyber threats. CTI aims to prepare for cyber threats and attacks in advance and share real-time information and minimize damage in the event of intrusions and attacks [3]. For effective use of CTI, individuals or organizations build sharing groups in peer-to-peer, peer-to-hub, or hybrid forms. Hence, even if an individual or an organization does not respond to all attacks independently, it is possible to collect attack patterns and learn how to respond by sharing attack information in the CTI sharing group [4].

The amount and accessibility of shared attack information are proportional to the size of the CTI group and its participants. In addition, research on a stable, scalable, and high-speed CTI deployment model is required to support a large-scale shared platform [5]. The Automated Indicator Sharing (AIS) is proposed to reduce the impact of cyberattacks by exchanging CTIs and defensive measures in real time between community participants [6]. To support the real-time exchange of CTI, AIS adopted a cyber threat information transmission standard TAXII (Trusted Automated eXchange of Indicator Information) [7] and a cyber threat information expression standard STIX (Structured Threat Information eXpression) [8].

TAXII is a transport standard that exchanges CTI through HTTP (HyperText Transport Protocol) [9] communication and operates in the application layer. Therefore, security in TAXII relies on HTTPS (HTTP Secure) [10]. Unfortunately, HTTPS cannot guarantee end-to-end security and privacy in resource-constrained network environments such as IoT using proxy devices. CTI may contain critical information about victims of cyber threats [11]. If the sharing platform cannot support complete end-to-end security and privacy, there is a risk of secondary damage to victims. Thus, additional application layer security for TAXII is required to prevent such unintended damages. In the CTI sharing platform using IoT, the use of EDHOC (Ephemeral Diffie-Hellman Over COSE) [12] together with lightweight IoT communication protocols such as CoAP (Constrained Application Protocol) [13] should be considered [14-15].

*Corresponding Author: Ilsun You; E-mail: ilsunu@gmail.com
DOI: 10.53106/160792642021092205011

EDHOC is currently the most promising yet to be the standardized approach to IoT communications security. Nevertheless, some threats have been disclosed by formally analyzing the protocol through different studies. For instance, authors in [16] and [17] formally verified EDHOC and analyzed diverse vulnerabilities that could violate the secrecy of the protocol. Hence it is imperative to remediate these vulnerabilities before the protocol is used to secure the IoT CTI sharing process [18]. Motivated by this, we propose an end-to-end privacy-preserving security protocol that works in two phases and improves the existing draft EDHOC protocol through priorly shared keys and key materials as well as ticket. The following are the main contributions of the paper:

- We design an end-to-end privacy-preserving security protocol that consists of the initial and re-authentication phases to configure the secure communication channel among the CTI sharing group.
- We prove the mutual authentication, secure key exchange, confidentiality, integrity, and anonymity properties of the proposed protocol using formal verification approaches.
- We comparatively analyze the proposed protocol against the draft EDHOC protocol regarding message overhead and computational and network latency.

Through the formal security verifications, the proposed protocol is proved to satisfy vital security requirements such as mutual authentication, secure key exchange, confidentiality, integrity, and anonymity. Furthermore, the experimental results illustrate that the proposed protocol, despite its larger message size and communication latency during initial authentication, achieves a better performance (compared to the draft EDHOC protocol) when there is frequent connection.

The rest of this paper is structured as follows: Section 2 describes the related works. Section 3 presents the threat models and vital security requirements required for the proposed protocol, which is described in detail by Section 4. The proposed protocol is validated with formal verification tools and is compared with the EDHOC protocol through experimental analysis in Sections 5 and 6, respectively. Finally, Section 7 concludes the paper.

2 Related Works

Today, especially in connection with 5G's promise to support Massive Machine Type Communications (mMTC), the IoT has become enormously pervasive to revolutionize many application areas such as vehicular communication, healthcare, cities, factories, Etc. However, the other side is the extended attack surface that stimulates hackers to steal and poison a great deal of private information. Like antivirus and firewalls, several reactive measures are already in place to defend against such information breaches, which are primarily suitable for known attacks. Hence, as new attack strategies and novel techniques and tools are devised, reactive measures come short of providing protections to the computing resources.

Attributable to the significant leaps in proficient methods and tools to realize artificial intelligence, it is now possible to effectively deploy proactive cybersecurity measures to mitigate threats that would not be possible otherwise. CTI serves this purpose by providing "evidence-based knowledge, including context, mechanisms, indicators, implications, and action-oriented advice about an existing or emerging menace or hazard to assets" [19]. For CTI to function the way it is intended, it is vital to collect various threat information from different sources like internal (network events, system logs, forensics, Etc.) and external (published vulnerabilities, newsfeeds, dark web, Etc.) [20]. A closely related and crucial aspect of CTI is threat intelligence sharing among cooperating participants to create cognizance of current vulnerabilities and threats. With this regard, various research has been conducted on the requirement of standard CTI sharing format for interoperability (such as STIX [8]), CTI transport mechanisms (such as TAXII [7]), security and privacy schemes [21], and laws and regulations governing CTI sharing [4].

Apart from collecting and sharing cyber threats for CTI, it is vital to ensure that these threats are accessible in a well-thought-out manner to both the human and the machine. To achieve this, STIX and TAXII are the two most commonly employed mechanisms- initially proposed by the US Department of Homeland Security (DHS) and are currently maintained by OASIS [22]. STIX aims to bring the interoperability of threat information shared among multiple stakeholders by devising a common language (using a graph-based model) and serialization (through JSON) standard. STIX is a graph-based language composed of nodes and edges. The nodes are defined by STIX Domain Objects and STIX Cyber-observable Objects, while STIX relationships define the edges. These objects are serialized using JSON encoded with UTF-8 as a mandatory to implement for STIX 2.1, although other serializations can also be used [8]. Once the threat information is collected and represented as STIX objects, they need to be transported efficiently and securely. That is where TAXII comes into the picture. It is an application layer protocol that is mainly designed to convey CTI over HTTPS [7]. CTI sharing through TAXII adheres to two modes of communications: collections and channels. The former specifies a service that enables consumers (TAXII clients) to send or receive information from a producer (TAXII server) as a request-response model. The latter, on the other hand, assists a message exchange among consumers in a publish/subscribe paradigm. That is, a producer can

receive messages from the set of TAXI clients as publishers and make available the information for consumption by authorized clients connected to the channel.

TAXII uses an HTTP protocol that works with a transport layer security (TLS) as a secure means to share CTI among cooperating organizations. Despite its robust features such as authenticated key exchange (AKE), forward secrecy for the long-term key, identity protection, and resilience against key compromise impersonation [23], (D)TLS is not a suitable or sufficient option for IoT communications for two main reasons. Primarily, (D)TLS is a heavy protocol that may consume a significant power of resource constrained IoT devices. For instance, a specific number of bytes in a DTLS 1.3 handshake performed via Elliptic Curve Diffie-Hellman Exchange (ECDHE) can be six times higher than the lightweight protocols proposed by The Internet Engineering Task Force (IETF) [12]. Next, given that IoT communications are frequently assisted with middleboxes, proxies, or gateways, they serve as endpoints that terminate TLS sessions. The fundamental problem with such arrangements is a deterioration of security as hop-by-hop security introduces additional threats to IoT cyberspace [24]. Another approach to the former challenge is to design a new lightweight application layer protocol called Constrained Application Protocol (CoAP) [13]. Besides its significantly lower overhead, the protocol supports multicast and request/response and publish/subscribe architectural models. The second problem can be addressed by putting forward an application layer protocol that maintains the broken end-to-end security, despite the presence of proxies and gateways. Object Security for Constrained RESTful Environments (OSCORE) [25] is such a protocol that serves this purpose by protecting CoAP messages. Like TLS uses TLS handshake protocol for key exchange between communicating parties, OSCORE uses a significantly lighter and squeezed key exchange protocol called Ephemeral Diffie-Hellman Over COSE (EDHOC) [12].

Apart from realizing desirable security properties such as mutual authentication, perfect forward secrecy, and identity protection, EDHOC optimizes the number of messages to be exchanged, the length of the message, and the number of encryptions, decryption and signing operations [16]. Unfortunately, EDHOC has various security issues identified by different authors such as [16] and [17]. The researchers mainly identified threats such as responder's identity disclosure, inability of the initiator to verify the responder's credential identity and unnecessarily prolonged EDHOC session during cipher suit rejection by the responder. Other significant security issues include assaults on the initiator's privacy and resource depletion attacks on the responder. As a result, to secure threat intelligence transmitted via TAXII in IoT settings, it is critical to enhance the EDHOC protocol in order to meet important security criteria while remaining lightweight.

3 Threat Model and Security Requirements

3.1 Intruder Model

Most security protocols are aimed to assist communicating peers to securely send and receive messages over insecure channels. For this to become reality, it is imperative for the protocols to defend themselves against various passive and active attacks. While passive attackers violate the confidentiality property of a system by observing the messages, active adversaries infringe the integrity and availability of the communication system by modifying (and deleting) the content of the messages. Furthermore, in particular to our protocol, we assume these intruders can also be insiders or unauthorized outsiders.

Security protocols that work in such burdensome environments should be modelled in such a way that they are aware of the intruders' capacity so that they can better defend attacks targeting them. The most suitable and most frequently used scheme for this purpose is the Dolev-Yao (DY) threat model [26]. The DY attacker is one of the strongest adversaries that is capable of eavesdropping on all messages transmitted, possesses its own copy of the authentication protocol, send/receive messages to/from communicating parties, forge/replay/delete messages and disrupt communication. In short, the DY attacker controls the transmission channel with limited exclusions. These exceptions are inability to encrypt or decrypt messages without possessing the correct keys, guess random numbers (nonce), reverse one-way hash functions, and solve the elliptic curve discrete logarithm problem.

Consequently, our proposed protocol is aimed to provide core security requirements such as mutual authentication, secure key exchange, perfect forward secrecy, privacy protection, confidentiality, and integrity despite the existence of the DY intruder.

3.2 Security Requirements

The main objective of the proposed protocol is to secure CTI sharing process in IoT environments. It is vital to protect CTI information as it contains critical evidences concerning cyberattacks (such as information about zero-day assaults that have yet to be reported). Therefore, when CTI producers share threat intelligence information, the proposed protocol must meet several security requirements, as described below.

- Mutual Authentication (Two-way Authentication): is an essential security property that dictates the Initiator and Responder to prove their identities before the actual communication proceeds.
- Secure Key Exchange: for the protocol to correctly

encrypt and integrity protect CTI messages, secure exchange of different keys (for instance pre-shared keys, ephemeral public keys and ephemeral session keys) is vital.

- Perfect Forward Secrecy: the proposed protocol guarantees to protect session keys by generating ephemeral keys for each session despite the cases when the long-term private keys of communicating parties are compromised.
- Privacy Protection: the proposed protocol keeps the identities of the Initiator and the Responder private by employing an Anonymity Identity (AID).
- Integrity: the keys and keying materials transmitted between the Initiator and the Responder should be protected from unauthorized modification. The proposed protocol realizes this by implementing hashing (and hash based message authentication codes).
- Confidentiality: similar to the Integrity property, all vital information sent and received is kept confidential by using an authenticated encryption with associated data (AEAD).

4 Proposed Protocol

4.1 Preliminaries

Table 1 outlines the notations used in the proposed protocol. According to the draft standard of EDHOC protocol, the Initiator always sends the first message in plain text, which exposes the EAD_1, ID_CRED_R, EAD_2, and other information[1] to an attacker, as they are sent to an unauthenticated party [12]. In the protocol we proposed; however, we use a temporary pre-shared key tPSK and an anonymous identity AID_0 that the Responder shares with the Initiator out of band. For this, the Responder primarily generates and stores $tPSK$ and ID_{tPSK}. It then uses its symmetric key K_R to encrypt the ID_{tPSK} together with a counter (the initial counter *count* is randomly generated and stored by Responder) to construct AID, as shown in equation (1). Scenarios like CTI sharing in IoT environments can perfectly suit such setups.

$$AID_i = E(K_R, ID_{tPSK} \mid count + i) \quad \textbf{(1)}$$

Having the shared key, the Initiator can now send the first message encrypted with tPSK so that the Responder identifies the sender and prevents an attacker from an authenticated Elliptic Curve Diffie-Helman (ECDH) key exchange and session key generation. It is also intended to protect the sender's identity from attackers by replacing the ID_{tPSK} with an AID. Furthermore, the tPSK and AID can also reset keys to proceed with Initial Authentication when the ticket expires. On the other hand, it is assumed that each party owns its ECDSA public key pair and the corresponding credential such as public key certificate.

In the next subsections, details of the steps involved in both phases (Initial Authentication and Re-authentication) of the proposed protocol are presented.

Table 1. Notations

Notation	Description
I	Initiator or its ECDSA private key
R	Responder or its ECDSA private key
PU_x	ECDSA Public key of X
AID_n	n^{th} Anonymous Identity
K_x	Secret Key of X
ID_{tPSK}	Identifier of the tPSK
Tpsk	temporal Pre-shared Key
PSK	Pre-shared Key
C_x	Session Identifier of X
G	The generator (basepoint)
x, y	ECDH private keys
$x \cdot G, y \cdot G$	ECDH public keys
$x \cdot y \cdot G$	ECDH session key
TYPE	EDHOC key type
$Corr$	EDHOC correlation
$SUITES_x$	EDHOC Suites of X
$CRED_x$	Credential of X
ID_CRED_x	Identifier of X's Credential
T(X)	X's ticket
Expired	Expiry time of T(X)
Seq	Sequence value of T(X)
AEAD(K; M)	Authenticated encryption with associated data
HMAC(K; M)	Hash-based message authentication code
HKDF()	simple key derivation function based on HMAC
|	concatenation

4.2 Initial Authentication

The "Initial Authentication" phase is based on the existing EDHOC procedure, where the Initiator and Responder perform mutual authentication and key exchange. Unique to our proposed protocol, the Responder provides the Initiator a pre-shared key PSK and Authentication Ticket T(I) for the first message protection and efficiency in the subsequent re-authentication phase. Details of this procedure are described as follows and shown in Figure 1.

(1) The Initiator performs the initial steps of preparing Msg1. First, it selects one from the four method types supported by EDHOC protocol, chooses the correlation value *corr*, prepares a list of cipher suites (in descending order of preference), and picks a connection identifier C_I. Next, it generates its ephemeral key pair, x and $x \cdot G$, and computes TYPE. Subsequently, it uses an AEAD encryption algorithm to encrypt $data_1$ with Tpsk, where $data_1$ consists of TYPE, $SUITES_I$, $x \cdot G$, C_I, and AID_i where the initial value of i is 0. Finally, it constructs Msg1 by appending AID_i with the cipher and sends that message to the Responder.

[1] For details, refer to [12]

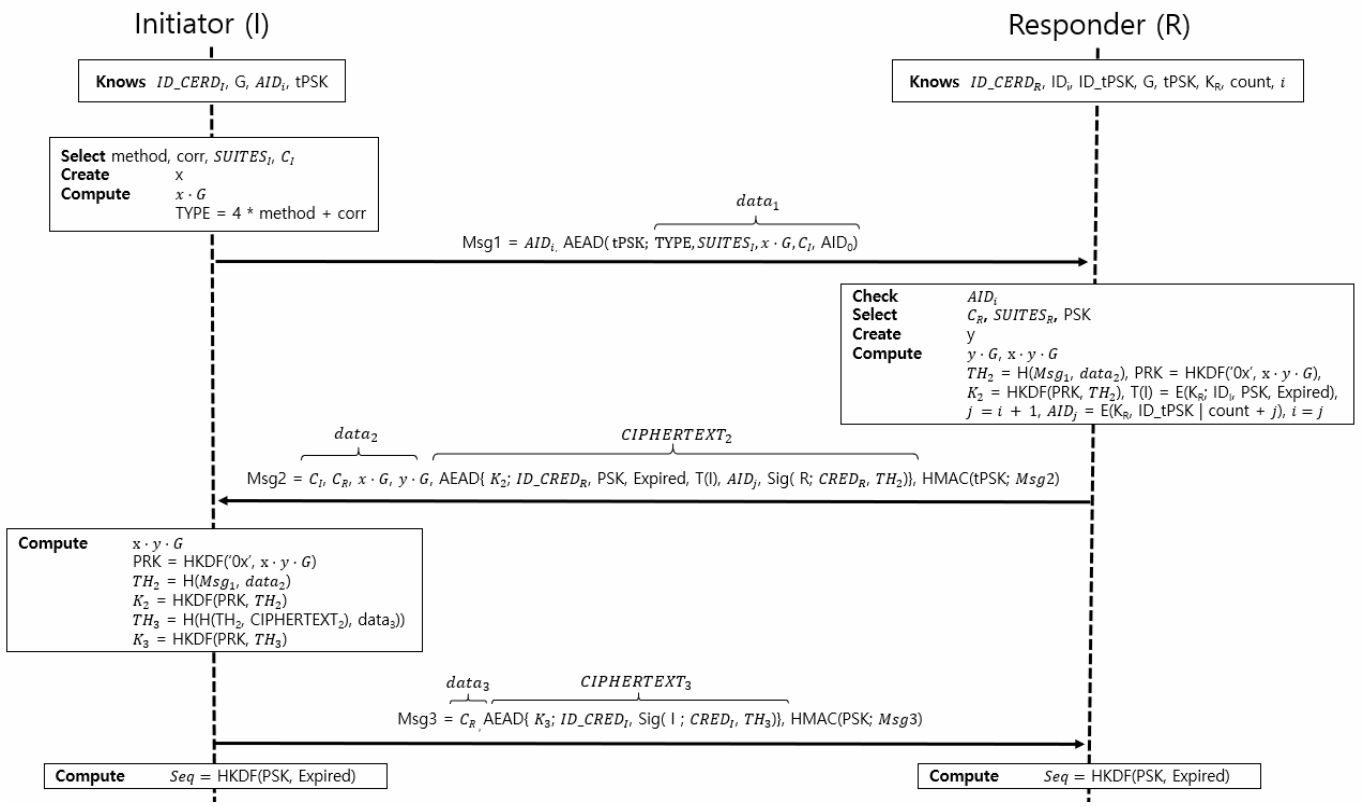

Figure 1. Initial Authentication phase

(2) When receiving Msg1, the Responder decrypts AID_i using its secret key K_R, followed by verifying the decrypted value (count+i) with its stored values. If this verification is successful, it retrieves the ID_{tPSK} mapping to a particular tPSK, hence being ready for validating the sender as a legitimate Initiator. Next, tPSK is used to decrypt the encrypted part to $data_1$. At this point, the Responder authenticates Msg1 by comparing the received AID_i with the decrypted AID_i. Only in the positive case, it proceeds the rest of this step, which includes costly public key operations, thereby thwarting resource exhaustion attacks. Successful authentication triggers the Responder to prepare C_R and $SUITES_R$ together with the temporary key pair (y and $y \cdot G$) and the ECDH shared secret $x \cdot y \cdot G$. It then computes the pseudo-random key PRK = HKDF('0x', $x \cdot y \cdot G$) and the transaction hash TH_2 = H(H(Msg1), $data_2$), where $data_2$ consists of C_I, C_R, $x \cdot G$, and $y \cdot G$. With PRK, it calculates the AEAD encryption key K_2 as HKDF(PRK, TH_2). Afterward, it signs the public credential $CRED_R$ and TH_2 with its ECDSA private authentication key R, followed by computing the ticket T(I) = E(K_R, ID_I, PSK, Expired) to be used in the Re-authentication phase and the next anonymous identifier AID_j = E(K_R, ID_{tPSK}|count + j) to be used in the new Initial Authentication. At this point, j = i+1, and after AID_j is computed, the current i is set to be j and stored. Finally, the Responder sends the Initiator Msg2 consisting of $data_2$ and $CIPHERTEXT_2$ with HMAC(tPSK, Msg2).

(3) On arrival of the message, the Initiator verifies the accompanied HMAC value with tPSK. Only if the validity holds, K_2 is calculated through ECDH the same way the Responder did. It then decrypts $CIPHERTEXT_2$ and stores PSK, Expired, T(I), and AID_j. Afterwards, it not only verifies the signature, but also calculates TH_3 as H(H(TH_2, $CIPHERTEXT_2$), $data_3$), where $data_3$ is C_R. Note that the Responder is here strongly authenticated to the Initiator based on the signature. The Initiator in turn computes the AEAD encryption key K_3 = HKDF(PRK, TH_3), signs the $CRED_I$ and TH_3 with its ECDSA private key I, and prepares $CIPHERTEXT_3$ by encrypting ID_CRED_I and the signature with K_3. Finally, it sends Msg3, composed of $data_3$ and $CIPHERTEXT_3$, with its HMAC value back to the Responder.

(4) When the Responder receives Msg3, it initially validates the accompanied HMAC value with PSK. On successful validation, it can confirm that the sender is a legitimate Initiator. The Responder then computes K_3 like the Initiator and decrypts $CIPHERTEXT_3$ to verify the Initiator's identity through ID_CRED_I. The positive verification shows that mutual authentication and secure key exchange are successful.

(5) Finally, the initiator and responder generate and store a sequence value Seq = HKDF(PSK, Expired) to prevent a replay attack on the initial message sent during the re-authentication phase.

4.3 Re-authentication

After the initial authentication, a "Re-authentication" phase can enable faster key exchange with less

computational overhead through authentication tickets and PSK. It is designed based on EDHOC's Symmetric key option, which is no longer included in the current draft standard. The Symmetric method has a vulnerability in which the identity of the Initiator is exposed by sending since ID_PSK is sent in plain. To address this issue, the protocol includes the Initiator's identifier ID_R in the authentication ticket, which is encrypted with K_R that is only known to the Responder. In addition, since the CTI message is transmitted immediately after the key exchange, the roundtrip is reduced from 1.5 to 1 by omitting the key confirmation process. Accordingly, network latency during the key exchange is reduced to support faster CTI sharing.

(1) As the first step, the Initiator encrypts $data_1$, consisting of TYPE, $SUITES_R$, $x \cdot G$, C_I, T(I) and *Seq*, with PSK. Note that T(I) and PSK are obtained in Msg2 of the "Initial Authentication." It then increments the value of *Seq* and sends Msg1 to the Responder to begin the re-authentication procedure.

(2) After decrypting Msg1 using K_R, the Responder checks if the decrypted *Seq* matches the one it possesses and the ticket T(I) has expired. In positive case indicating the Initiator is successfully authenticated, it generates the ECDH key pair $(y \cdot y \cdot G)$, computes $x \cdot y \cdot G$, and composes $data_2$. Next, it computes the AEAD encryption key K_2 in the same manner that it did in the first authentication phase, except that PRK is now computed as HKDF(PSK, $x \cdot y \cdot G$). Afterwards, as shown in Figure 2, the Responder creates a new message, Msg2, which contains $data_2$ and $CIPHERTEXT_2$ and transmits it together with its HMAC value to the Initiator. Finally, the Responder computes the session key K_3 as HKDF(PRK, TH_3) and increments *Seq* by one.

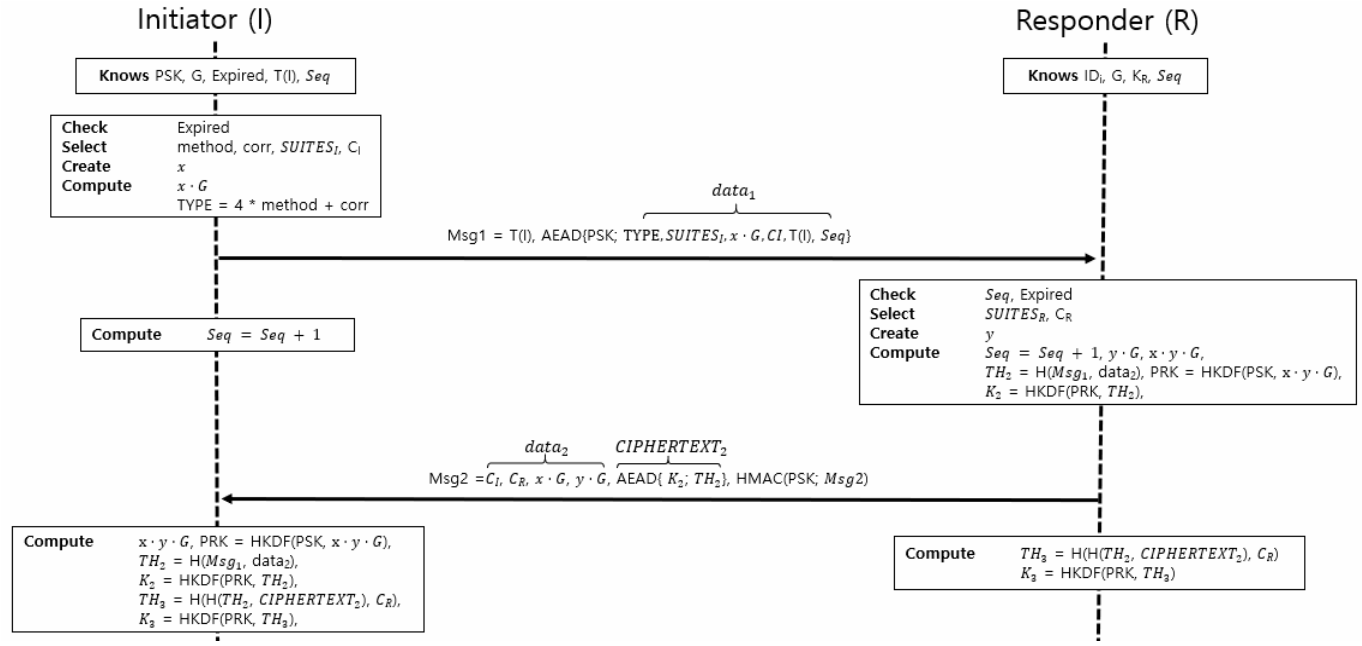

Figure 2. Re-authentication phase

(3) On arrival of Msg2, the Initiator validates the HMAC value of Msg2. If successful, it computes $x \cdot y \cdot G$ to derive PRK and K_2 and then recovers TH_2 by decrypting $CIPHERTEXT_2$. It in turn calculates TH_3 and PRK to get the session key K_3 and immediately initiates protected application-layer communication.

5 Formal Verification

5.1 Formal Verification with BAN-logic

In this section, the formal security analysis of both phases of the protocol using BAN-logic [27] and AVISPA [28] is carried out. By applying a formal verification, the security of the proposed protocol can be confirmed and guaranteed.

BAN-logic is a modal-logic-based formal verification tool that analyzes authentication protocols through a series of steps: (1) Idealization, (2) Assumption, (3) Goal, and (4) Derivation.

In the first step, all unprotected messages transmitted over the insecure channel are excluded, and only those that are ciphered (e.g., encrypted messages, hash-based message authentication codes, and digital signatures) are idealized. Once the message flows are represented in idealized form, suitable Assumptions and Goals are defined. Finally, the goals are derived from the other three steps and the BAN-logic rules. Tables 2 and Table 3 summarize the symbols and formulas used in the BAN-logic verification process, respectively.

Table 2. BAN-logic notations

Notation	Meaning
$P \mid\equiv X$	P believes that the message X is true
$P \lhd X$	P receives the message X at any point in time
$P \mid\sim X$	P previously sent the message X
$P \mid\Rightarrow X$	P has jurisdiction over X
$\#(X)$	X is fresh
$P \overset{x}{\leftrightarrow} Q$	K is a secret key shared between P and Q
$P \overset{x}{\Leftrightarrow} Q$	K is a shared secret between P and Q
$\{X\}_x$	X is encrypted with a key K
X, Y	X is combined with Y

Table 3. BAN-logic rules

Rules	Description
Message Meaning Rule (MM)	$\frac{P \mid\equiv P \overset{K}{\leftrightarrow} Q, P \lhd \{X\}_K}{P \mid\equiv Q \mid\sim X}$ $\frac{P \mid\equiv P \overset{K}{\Leftrightarrow} Q, P \lhd \langle X \rangle_K}{P \mid\equiv Q \mid\sim X}$
Nonce Verification Rule (NV)	$\frac{\boldsymbol{P} \mid\equiv \overset{\boldsymbol{K}}{\rightarrow} \boldsymbol{Q}, \boldsymbol{P} \lhd \{\boldsymbol{X}\}_{\boldsymbol{K}^{-1}}}{\boldsymbol{P} \mid\equiv \boldsymbol{Q} \mid\sim \boldsymbol{X}}$ $\frac{\boldsymbol{P} \mid\equiv \#(\{\boldsymbol{X}\}), \boldsymbol{P} \mid\equiv \boldsymbol{Q} \sim \boldsymbol{X}}{\boldsymbol{P} \mid\equiv \boldsymbol{Q} \mid\equiv \boldsymbol{X}}$
Jurisdiction Rule (JR)	$\frac{\boldsymbol{P} \mid\equiv \boldsymbol{Q} \Rightarrow \boldsymbol{X}, \boldsymbol{P} \mid\equiv \boldsymbol{Q} \mid\equiv \boldsymbol{X}}{\boldsymbol{P} \mid\equiv \boldsymbol{X}}$
Freshness Rule (FR)	$\frac{\boldsymbol{P} \mid\equiv \#(\boldsymbol{X})}{\boldsymbol{P} \mid\equiv \#(\boldsymbol{X}, \boldsymbol{Y})}$
Decomposition Rule (DR)	$\frac{\boldsymbol{P} \lhd (\boldsymbol{X}, \boldsymbol{Y})}{\boldsymbol{P} \lhd \boldsymbol{X}}$ $\frac{P \mid\equiv X, P \mid\equiv Y}{P \mid\equiv (X, Y)}$
Belief Conjunction Rule (BC)	$\frac{P \mid\equiv Q \mid\equiv (X, Y)}{P \mid\equiv Q \mid\equiv X}$ $\frac{\boldsymbol{P} \mid\equiv \boldsymbol{Q} \mid\sim (\boldsymbol{X}, \boldsymbol{Y})}{\boldsymbol{P} \mid\equiv \boldsymbol{Q} \sim \boldsymbol{X}}$
Diffie-Hellman Rule (DH)	$\frac{P \mid\equiv Q \mid\sim \xrightarrow{d2 \cdot G} Q, P \mid\equiv \xrightarrow{d1 \cdot G} P}{P \mid\equiv P \xrightarrow{d1 \cdot d2 \cdot G} Q}$

For the initial authentication stage, the idealized forms of the messages are illustrated as (II1)- (II3). Thereafter, the assumptions regarding freshness property (IA2, IA5, IA7, IA10, IA12, IA13, and IA15), ephemeral Diffie-Hellman public keys (IA3 and IA6), symmetric keys (IA1, IA4, IA8, and IA11), and public keys (IA9 and IA14) are set. Concerning the goals, (IG8) and (IG11) demonstrate mutual authentication; (IG2), (IG3), (IG4) and (IG10) represent perfect forward secrecy; (G1) and (G7) show Anonymity; and (IG5), (IG6) and (IG9) denote secure key exchange. According to the derivation results, all the security requirements described in subsection 3.2 are satisfied. Figure 3 shows the BAN-logic formal security verification of the initial authentication phase.

As to the re-authentication phase, the BAN-logic verification steps are shown in Figure 4. In this phase, the assumptions are basically regarding freshness (RA2, RA4, RA7, and RA9), the ephemeral Diffie-Hellman public keys (RA5 and RA8), and the pre-shared symmetric key (RA1, RA3, and RA6). Concerning the goals, (RG2) and (RG5) demonstrate mutual authentication; (RG4), (RG7), and (RG8) represent perfect forward secrecy; and (RG1), (RG3), and (RG6) denote secure key exchange. Table 4 maps the security properties to the BAN-logic derivations of both phases of the proposed protocol.

In conclusion, both phases of the proposed protocol satisfy SP1 to SP5 (mutual authentication, secure key exchange, perfect forward secrecy, confidentiality, and integrity). In addition, the initial authentication phase of the proposed protocol fulfills SP6 (Anonymity). In the following subsection, the same protocol is analyzed using the AVISPA tool.

5.2 Formal Verification with AVISPA

It is often essential to not only rely on the results of one verification tool, in particular to our case, on BAN-Logic only [29]. Having at least two formal verification approaches to a single protocol enables one to complement the weakness of the other and make the outcome of verification stronger. Accordingly, we use the AVISPA tool as a secondary verification mechanism. It is an automated validation tool for modeling and analyzing different authentication protocols. To verify a security protocol with AVISPA, the security protocol must be modeled in a specification language called HLPSL [30], which is then converted to the intermediate format (IF) via the HLPSL2IF component. The IF model then passes through the four backend modules: On-the-Fly Model Checker (OFMC) [31], CL-based Attacker Searcher (CL-AtSe) [32], SAT-based Model-Checker (SATMC) [33], and Tree Automata-based Protocol Analyzer (TA4SP) [34]. Finally, the verification result comes out in output format (OF), as depicted in Figure 5.

At first, both phases of the proposed protocol are modeled in HLPSL code. HLPSL codes are generally divided into roles such as basic role, session role, and environment role. The basic role expresses the basic behavior and transport of the protocol participant. In the session role, the agents and other parameters used in the basic role are defined. Finally, the environment role contains global constants, knowledge of intruders, a parallel configuration of sessions, and verification goals. HLPSL codes of the proposed protocol are conducted with basic roles of the Initiator I and the Responder R, session, and environment. Table 5 shows the goals that the proposed protocol is expected to satisfy and that AVISPA verifies. AVISPA tool checks

Figure 3. BAN-logic based formal verification of the initial stage

Table 4. Security property satisfaction

No.	Security Properties		Initial Authentication	Re-authentication
SP1	Mutual authentication		(ID17), (ID26)	(RD7), (RD13)
SP2	Secure key exchange	I	(ID13), (ID14)	(RD4), (RD8)
		R	(ID21), (IA11)	(RD14), (RA6)
SP3	Perfect forward secrecy		(ID5), (ID9)	(RD9), (RD15)
SP4	Confidentiality		SP2, SP3	SP2, SP3
SP5	Integrity		SP2, SP3	SP2, SP3
SP6	Anonymity		(ID4), (ID15)	--

(1) Idealization

$(RI1)\ I \rightarrow R: T(I), \left\{Data_1, I \overset{PSK}{\longleftrightarrow} R\right\}_{PSK}$

where $T(I) = \left\{I \overset{PSK}{\longleftrightarrow} R, Expired\right\}_{K_R}, Data_1 = \left[TYPE, SUITES_I, \overset{d1\cdot G}{\longrightarrow} I, C_I, Seq, T(I)\right]$

$(RI2)\ R \rightarrow I: \left\{Data_2, CIPHERTEXT_2, I \overset{PSK}{\longleftrightarrow} R\right\}_{PSK}$

where $Data2 = \left[C_I, C_R, \overset{d1\cdot G}{\longrightarrow} I, \overset{d2\cdot G}{\longrightarrow} R\right], CIPHERTEXT_2 = \left\{TH2, I \overset{d1\cdot d2\cdot G}{\longleftrightarrow} R\right\}_{\overset{d1\cdot d2\cdot G}{I \longleftrightarrow R}}$

(2) Assumption

$(RA1)\ R \mid \equiv R \overset{K_R}{\longleftrightarrow} R$

$(RA2)\ R \mid \equiv \#(Expired)$

$(RA3)\ R \mid \equiv R \mid \Rightarrow I \overset{PSK}{\longleftrightarrow} R$

$(RA4)\ R \mid \equiv \#(Seq)$

$(RA5)\ R \mid \equiv \overset{d2\cdot G}{\longrightarrow} R$

$(RA6)\ I \mid \equiv I \overset{PSK}{\longleftrightarrow} R$

$(RA7)\ I \mid \equiv \#\left(\overset{d1\cdot G}{\longrightarrow} I\right)$

$(RA8)\ I \mid \equiv \overset{d1\cdot G}{\longrightarrow} I$

$(RA9)\ I \mid \equiv \#\left(I \overset{d1\cdot d2\cdot G}{\longleftrightarrow} R\right)$

(3) Goal

$(RG1)\ R \mid \equiv I \overset{PSK}{\longleftrightarrow} R$

$(RG2)\ R \mid \equiv I \mid \equiv I \overset{PSK}{\longleftrightarrow} R$

$(RG3)\ R \mid \equiv I \overset{d1\cdot d2\cdot G}{\longleftrightarrow} R$

$(RG4)\ I \mid \equiv R \mid \equiv I \overset{PSK}{\longleftrightarrow} R$

$(RG5)\ I \mid \equiv I \overset{d1\cdot d2\cdot G}{\longleftrightarrow} R$

$(RG6)\ I \mid \equiv R \mid \equiv I \overset{d1\cdot d2\cdot G}{\longleftrightarrow} R$

(4) Derivation

$(RD1)\ R \triangleleft T(I), \left\{Data_1, I \overset{PSK}{\longleftrightarrow} R\right\}_{PSK}$

$(RD2)\ R \mid \equiv R \mid\sim \left[I \overset{PSK}{\longleftrightarrow} R, Expired\right]$ by $(RD1), DC, (RA1), MM$

$(RD3)\ R \mid \equiv R \mid \equiv \left[I \overset{PSK}{\longleftrightarrow} R, Expired\right]$ by $(RD2), (RA2), FR, NV$

$(RD4)\ R \mid \equiv I \overset{PSK}{\longleftrightarrow} R$ by $(RD3), BC, (RA3), JR$

$(RD5)\ R \mid \equiv I \mid\sim \left[Data_1, I \overset{PSK}{\longleftrightarrow} R\right]$ by $(RD1), DC, (RD4), MM$

$(RD6)\ R \mid \equiv I \mid \equiv \left[Data_1, I \overset{PSK}{\longleftrightarrow} R\right]$ by $(RD5), (RA4), FR, NV$

$(RD7)\ R \mid \equiv I \mid \equiv I \overset{PSK}{\longleftrightarrow} R$ by $(RD6), BC$

$(RD8)\ R \mid \equiv I \overset{d1\cdot d2\cdot G}{\longleftrightarrow} R$ by $(RD6), BC, (RA5), DH$

$(RD9)\ I \triangleleft \left\{Data_2, CIPHERTEXT_2, I \overset{PSK}{\longleftrightarrow} R\right\}_{PSK}$

$(RD10)\ I \mid \equiv R \mid\sim \left[Data_2, CIPHERTEXT_2, I \overset{PSK}{\longleftrightarrow} R\right]$ by $(RD9), (RA6), MM$

$(RD11)\ I \mid \equiv R \mid \equiv \left[Data_2, CIPHERTEXT_2, I \overset{PSK}{\longleftrightarrow} R\right]$ by $(RD10), (RA7), FR, NV$

$(RD12)\ I \mid \equiv R \mid \equiv I \overset{PSK}{\longleftrightarrow} R$ by $(RD11), BC$

$(RD13)\ I \mid \equiv I \overset{d1\cdot d2\cdot G}{\longleftrightarrow} R$ by $(RD10), BC, (RA8), DH$

$(RD14)\ I \mid \equiv R \mid\sim \left[TH2, I \overset{d1\cdot d2\cdot G}{\longleftrightarrow} R\right]$ by $(RD11), BC, (RD13), MM$

$(RD15)\ I \mid \equiv R \mid \equiv \left[TH2, I \overset{d1\cdot d2\cdot G}{\longleftrightarrow} R\right]$ by $(RD14), (RA9), FR, NV$

$(RD16)\ I \mid \equiv R \mid \equiv I \overset{d1\cdot d2\cdot G}{\longleftrightarrow} R$ by $(RD15), BC$

Figure 4. BAN-logic based formal verification of the re-authentication stage

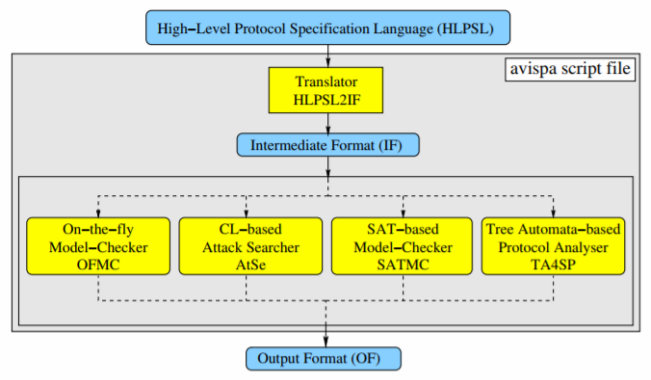

Figure 5. AVISPA structure

with two backend modules OFMC and CL-AtSe whether HLPSL codes of the proposed protocol satisfy the security properties and presents the verification result as shown in Figure 6 (initial phase) and Figure 7 (re-authentication phase), respectively.

Table 5. Additional codes for verification

Code	Meanings
secret (PSK', secret_psk, {A, B})	PSK should be confidential
secret (K2', secret_k2, {A, B})	K2 should be confidential
secret (K3', secret_k3, {A, B})	K3 should be confidential
witness (B, A, auth1, K2') /	Initiator authenticates
request (A, B, auth1, K2')	Responder with K2
witness (A, B, auth2, K3') /	Responder authenticates
request (B, A, auth2, K3')	Initiator with K3
witness (A, B, auth3, K3') /	Responder authenticates
request (B, A, auth3, K3')	Initiator with PSK
witness (B, A, auth4, K2') /	Initiator authenticates
request (A, B, auth4, K2')	Responder with K2

According to the Figure 6 and Figure 7, both phases of the proposed protocol are proven to be secure against known attacks. Consequently, the verification results from BAN-logic and AVISPA show that the proposed protocol is secure.

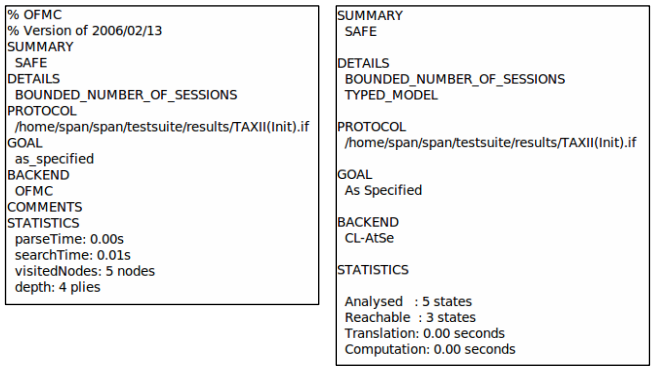

Figure 6. Verification outcome for the initial stage

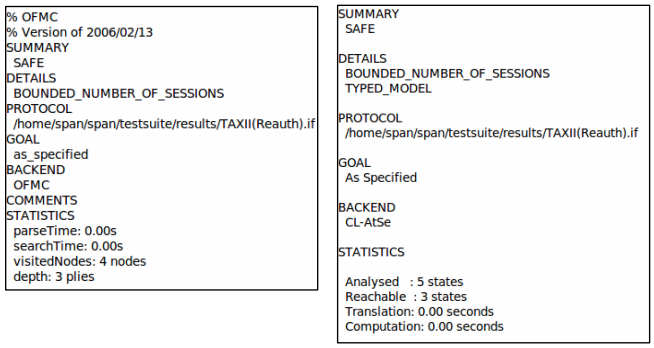

Figure 7. Verification outcome for re-authentication stage

6 Experimental Results

This section presents the experimental works carried out in implementing the proposed protocol and the EDHOC protocols. Each protocol is developed in C++ and implemented in the two entities Initiator and Responder with specifications shown in Table 6. In addition, the EDHOC cipher suites and additional encryption and hashing algorithms (for ticket, AID generation and integrity verification) are shown in Table 7. Based on these setups, we measure the computational and network latency of each protocol. In doing so, we follow the approach taken by Kodali et al. to construct an experimental environment using NS3 model [35].

In calculating the computational time overhead associated with each protocol, we measure the time required to create the ongoing and process the arriving messages. The incoming message processing time is essentially the time spent decoding the message. It also includes the processing time for the verification of hashed message authentication codes and digital signatures. To measure network latency, we build a simulation environment as previously described and measured the latency of each message by sending packets with equal size as shown in Table 8. Table 9 presents the simulation results. The total message lengths of the Initial Authentication and Re-authentication protocols are 400 bytes and 171 bytes, respectively. In addition, using the proposed protocol,

Table 6. Testbed specification for the Initiator and Responder

Item	Initiator	Responder
CPU	Broadcom BCM2711	Intel Core i5-6300HQ
RAM	2 GB	8GB
Compiler	g++ 8.3	Visual studio 2019
OS	Raspberry Pi OS	Windows 10 64bit

Table 7. EDHOC SUITES and Algorithms in Tests

Algorithm	Description
EDHOC AEAD	AES-CCM-16-64-128
EDHOC HASH	SHA-256
EDHOC ECDH CURVE	X25519
EDHOC SIGN	ECDSA
EDHOC SIGN CURVE	ED25519
Ticket, AID Encryption	AES-128-CTR
HMAC	HMAC-SHA256

Table 8. Message size

Protocols	Message size (bytes)			
	Msg1	Msg2	Msg3	Total
EDHOC	37	118	91	246
Initial Authentication	63	214	123	400
Re-authentication	105	66		171

Table 9. Computational and network latency

Protocols	latency (millisecond)			
	Msg1	Msg2	Msg3	Total
EDHOC	1	7	5	13
Initial Authentication	4	20	10	34
Re-authentication	3	5		8

Initiator and Responder took 34 ms for the initial authentication and 8 ms for the re-authentication. The previously reported measurements can vary in real-world network environments and may probably exhibit an increased computational time and network latency.

The experimental result explicates the proposed protocol has a higher message size hence a higher computational and network latency. However, the Initial Authentication protocol (which accounts for 70% of the total message length and 81% of the total network latency introduced) does not occur often. This makes the protocol efficient in the long run as we designed a re-authentication protocol with a single roundtrip to support frequent CTI sharing. Furthermore, because the Re-authentication protocol does not use digital signatures, it supports faster key exchange than the EDHOC protocol while ensuring faster authentication processing time. Figure 8 displays this fact by showing the trend that as number of sessions grow, both message overheads and network latencies significantly reduced. This improves the network performance by 62.5%.

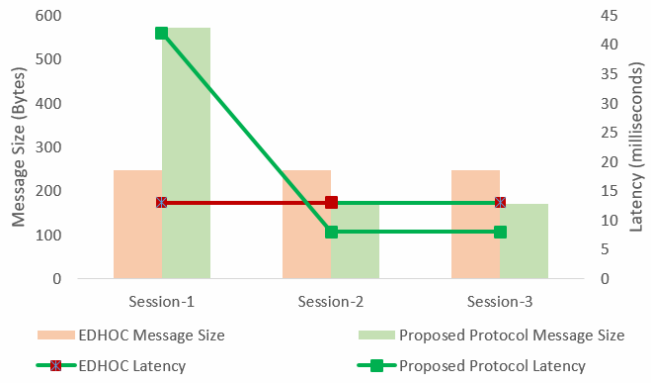

Figure 8. Comparison of EDHOC and the proposed protocol concerning message size and latency

7 Conclusion

CTI sharing has been one of the essential means of proactive defense of security attacks. At present, TAXII and STIX serve as the de facto methods to share intelligence among cooperating entities. CTI sharing currently utilizes HTTPS, which is inefficient in an IoT setting owing to its transparency and resource-intensive nature. Accordingly, we put forward a two-phase security protocol that improves the draft EDHOC protocol by using pre-shared keys and keying materials. In the initial authentication phase, the Responder supplies the Initiator with a pre-shared key PSK and an Authentication Ticket T(I) for initial message protection and fast re-authentication. Subsequently, the re-authentication phase enables quicker key exchange with reduced computing cost using the authentication tickets and PSK. The formal security verification of our protocol shows a positive result in satisfying vital security properties like mutual authentication, secure key exchange, perfect forward secrecy, anonymity, confidentiality, and integrity. Furthermore, comparing the protocol's performance to that of the EDHOC protocol reveals a substantial improvement with a single roundtrip to allow frequent CTI sharing.

Acknowledgements

This work was supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract (UD190016ED).

References

[1] V. Mavroeidis, S. Bromander, Cyber Threat Intelligence Model: An Evaluation of Taxonomies, Sharing Standards, and Ontologies within Cyber Threat Intelligence, *Proc. of the 2017 European Intelligence and Security Informatics Conference (EISIC'17)*, Athens, Greece, 2017, pp. 91-98.

[2] A. Abhishta, W. Heeswijk, M. Junger, L. Nieuwenhuis, R. Joosten, Why would we get attacked? An analysis of attacker's aims behind DDoS attacks, *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, Vol. 11, No. 2, pp. 3- 22, June, 2020.

[3] M. Alizadeh, K. Andersson, O. Schelen, 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, August, 2020.

[4] T. D. Wagner, K. Mahbub, E. Palomar, A. E. Abdallah, Cyber threat intelligence sharing: Survey and research directions, *Computers and Security*, Vol. 87, Article No. 101589, November, 2019.

[5] J. Heo, Y. E. Gebremariam, H. Park, B. Kim, I. You, Study on Hybrid Cloud-based Cyber Threat Intelligence Sharing Model Requirements Analysis, *Proc. of the 2020 ACM International Conference on Intelligent Computing and its Emerging Applications (ICEA'20)*, Gangwon, Republic of Korea, 2020.

[6] CISA, *Automated Indicator Sharing*, Available: https://www.cisa.gov/ais [Online; accessed on July 15, 2021].

[7] OASIS Cyber Threat Intelligence (CTI) TC, *TAXII™ version 2.1*, OASIS OPEN, OASIS Standard taxi-v2.1-os, pp. 1-79, June, 2021.

[8] OASIS Cyber Threat Intelligence (CTI) TC, *STIX™ version 2.1*, OASIS OPEN, OASIS Standard stix-v2.1-os, pp. 1-313, June, 2021.

[9] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, *Hypertext Transfer Protocol – HTTP/1.1*, IETF RFC 2616, June, 1999.

[10] E. Rescorla, *HTTP Over TLS*, IETF RFC 2818, May, 2000.

[11] S. K. Wong, S. M. Yiu, Location spoofing attack detection with pre-installed sensors in mobile devices, *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, Vol. 11, No. 4, pp. 16-30, December, 2020.

[12] G. Selander, J. Mattsson, F. Palombini, *Ephemeral Diffie-Hellman Over COSE (EDHOC)*, IETF Internet-Draft draft-ietf-lake-edhoc-08, July, 2021.

[13] Z. Shelby, K. Hartke, C. Bormann, *The Constrained Application Protocol (CoAP)*, IETF RFC 7252, June, 2014.

[14] J. C. Loh, S. H. Heng, S. Y. Tan, K. Kurosawa, On the Invisibility and Anonymity of Undeniable Signature Schemes, *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, Vol. 11, No. 1, pp. 18-34, March, 2020.

[15] S. Y. Moon, J. H. Park, J. H. Park, Authentications for Internet of Things Security: Threats, Challenges and Studies, *Journal of Internet Technology*, Vol. 19, No. 2, pp. 349-358, March, 2018.

[16] A. Bruni, T. S. Jørgensen, T. G. Petersen, C. Schürmann, Formal verification of ephemeral Diffie-Hellman over COSE (EDHOC), *Proc. of the 4th International Conference on Research in Security Standardisation (SSR'18)*, Darmstadt, Germany, 2018, pp. 21-36.

[17] K. Norrman, V. Sundararajan, A. Bruni, *Formal Analysis of EDHOC Key Establishment for Constrained IoT Devices*,

arXiv preprint arXiv:2007.11427, September, 2020.

[18] Y. Tian, Y. Li, X. Liu, R. H. Deng, B. Sengupta, Privacy-Preserving Biometric-Based Remote User Authentication, *Journal of Internet Technology*, Vol. 20, No. 7, pp. 2265-2276, December, 2019.

[19] R. McMillan, *Definition: Threat Intelligence*, Gartner Research, May, 2013.

[20] A. Ramsdale, S. Shiaeles, N. Kolokotronis, A comparative analysis of cyber-threat intelligence sources, formats and languages, *Electronics*, Vol. 9, No. 5, Article No. 824, May, 2020.

[21] D. Homan, I. Shiel, C. Thorpe, A new network model for cyber threat intelligence sharing using blockchain technology, *Proc. of the 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS'19)*, Canary Islands, Spain, 2019, pp. 1-6.

[22] OASIS, *OASIS Cyber Threat Intelligence (CTI) TC*, 2021. Available: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cti [Online; accessed on July 15, 2021]

[23] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, IETF RFC 8446, August, 2018.

[24] Open Mobile Alliance, *An Application-Layer Approach to End-to-End Security for the Internet of Things*, June, 2019. Available online: https://omaspecworks.org/end-to-end-security-for-the-internet-of-things/ [Online; accessed on July 15, 2021]

[25] G. Selander, J. Mattsson, F. Palombini, L. Seitz, *Object Security for Constrained RESTful Environments (OSCORE)*, IETF RFC 8613, July, 2019.

[26] D. Dolev, A. Yao, On the security of public key protocols, *IEEE Transactions on Information Theory*, Vol. 29, No. 2, pp. 198-208. March, 1983.

[27] M. Burrows, M. Abadi, R. M. Needham, A logic of authentication, *Proc. of The Royal Society of London. A. Mathematical and Physical Sciences*, Vol. 426, No. 1871, pp. 233-271, December, 1989.

[28] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P. C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganèo, L. Vigneron, The AVISPA tool for the automated validation of Internet security protocols and applications, *Proc. of the 17th International Conference on Computer Aided Verification (CAV'05)*, Edinburgh, Scotland, UK, 2005, pp. 281-285.

[29] C. Boyd, W. Mao, On a limitation of BAN logic, *Proc. of the 1993 Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT'93)*, Lofthus, Norway, 1993, pp. 240-247.

[30] The AVISPA team, *HLPSL tutorial*, Research Report IST-2001-39252, June, 2006.

[31] D. Basin, S. Mödersheim, L. Viganò, OFMC: A symbolic model checker for security protocols, *International Journal of Information Security*, Vol. 4, No. 3, pp. 181-208, June, 2005.

[32] M. Turuani, The CL-Atse Protocol Analyser, *Proc. of the 17th International Conference on Rewriting Techniques and Applications (RTA'06)*, Seattle, Washington, USA, 2006, pp. 277-286.

[33] A. Armando, L. Compagna, SATMC: A SAT-Based Model Checker for Security Protocols, *Proc. of the 9th European Workshop on Logics in Artificial Intelligence (JELIA'2004)*, Lisbon, Portugal, 2004, pp. 730-733.

[34] Y. Boichut, P.-C. Héam, O. Kouchnarenko, *Automatic Verification of Security Protocols Using Approximations*, Research Report RR-5727, INRIA, May, 2006.

[35] R. K. Kodali, B. Kirti, NS-3 Model of an IoT network, *Proc. of the 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA'20)*, Greater Noida, India, 2020, pp. 699-702.

Biographies

Hoonyong Park received the B.S. degree in Information Security Engineering from SoonCheonHyang University, Asan-si, South Korea, in 2019. He is currently a Integrated Ph.D. program student with Department of Information Security Engineering of SoonCheonHyang University, Asan-si, South Korea. His research topics are Formal security verification, 5G Networks.

Jiyoon Kim received the M.S. degree in information security engineering from Soonchunhyang University, Asan, South Korea, in 2019. He is currently pursuing the Ph.D. degree in Information Security Engineering from the same University. His current research interests include mobile Internet security, 5G security, and formal security analysis.

Sangmin Lee received the B.S. degree in Information Security Engineering from SoonCheonHyang University, Asan-si, South Korea, in 2020. She is currently a master's degree student with Department of Information Security Engineering of SoonCheonHyang University, Asan-si, South Korea. Her research topics are Formal security verification, 5G Networks and secure protocol.

Daniel Gerbi Duguma received the M.Sc. degree in Information Security Engineering from Soonchunhyang University, South Korea, in 2021, where he is currently pursuing the Ph.D. degree with the Department of Information Security Engineering. His research interests include 5G and beyond Security, IoT Security, and Formal Security Analysis.

Ilsun You received the M.S. and Ph.D. degrees in computer science from Dankook University, Seoul, South Korea, in 1997 and 2002, respectively, and the Ph.D. degree from Kyushu University, Japan, in 2012. He is currently working as a full professor at the Department of Information Security Engineering, Soonchunhyang University, South Korea. His main research interests include Internet security, authentication, access control, and formal security analysis.