IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 16, 2020, accepted June 25, 2020, date of publication June 29, 2020, date of current version July 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3005592

LAKS-NVT: Provably Secure and Lightweight Authentication and Key Agreement Scheme Without Verification Table in Medical Internet of Things

KISUNG PARK[1], SUNGKEE NOH[1], HYUNJIN LEE[1], ASHOK KUMAR DAS[2], (Senior Member, IEEE), MYEONGHYUN KIM[3], YOUNGHO PARK[3], (Member, IEEE), AND MOHAMMAD WAZID[4], (Senior Member, IEEE)

[1]Blockchain Technology Research Center, Electronics and Telecommunications Research Institute, Daejeon 34129, South Korea
[2]Center for Security, Theory, and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India
[3]School of Electronics Engineering, Kyungpook National University, Daegu 41566, South Korea
[4]Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun 248002, India

Corresponding author: Youngho Park (parkyh@knu.ac.kr)

This work was supported by the Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean Government (Research on Intelligent Cyber Security and Trust Infra) under Grant 20ZR1300.

ABSTRACT Wireless body area networks (WBANs) and wireless sensor networks (WSNs) are important concepts for the Internet of Things (IoT). They have been applied to various healthcare services to ensure that users can access convenient medical services by exchanging physiological data between user and medical server. User physiological data is collected by sensor nodes and sent to medical service providers, doctors, etc. using public channels. However, these channels are vulnerable to various potential attacks, and hence, it is essential to design provably secure and lightweight mutual authentication (MA) schemes for medical IoT to protect user privacy and achieve secure communication. A lightweight mutual authentication and key agreement (MAKA) scheme was designed in 2019 to guarantee user privacy, but we found that the scheme does not withstand impersonation, stolen senor node and leaking verification table attacks, and it does not also ensure anonymity, untraceability and secure mutual authentication. This paper proposes a provably secure and lightweight MAKA scheme for medical IoT, called LAKS Non-verification table (NVT), that does not require a server verification table. We assess LAKS-NVT's security against various potential attacks and demonstrate that it achieves secure MA between sensor node and server using Burrows-Abadi-Needham logic. We employ the well-known Real-Or-Random which is random oracle model to prove that LAKS-NVT provides a session key security. In addition, the formal security verification using the widely-accepted Automated Validation of Internet Security Protocols and Applications (AVISPA) software tool has been performed and the results show that LAKS-NVT is also secure. We compare LAKS-NVT's performance against contemporary authentication schemes, and verify that it achieves better security and comparable efficiency. The practical perspective of LAKS-NVT is also carried out via the Network Simulator 2 (NS2) simulation study.

INDEX TERMS Authentication, key agreement, medical Internet of Things, NS2 simulation, ROR model, session key security.

I. INTRODUCTION

A. BACKGROUND AND MOTIVATION

Recent information and communication (ICT) and embedded technology advances have facilitated the emerging internet of things (IoT) development. IoT will include over 50 billion devices linked to the internet by 2020, with users employing a variety of convenient services based on IoT devices, such as smart homes, smart-cities, smart health care, smart grid, etc. [1]. Medical IoT, i.e., wireless body area networks (WBANs), and health care services are particularly important IoT components focused on improving human quality of human.

The associate editor coordinating the review of this manuscript and approving it for publication was Lorenzo Mucchi.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

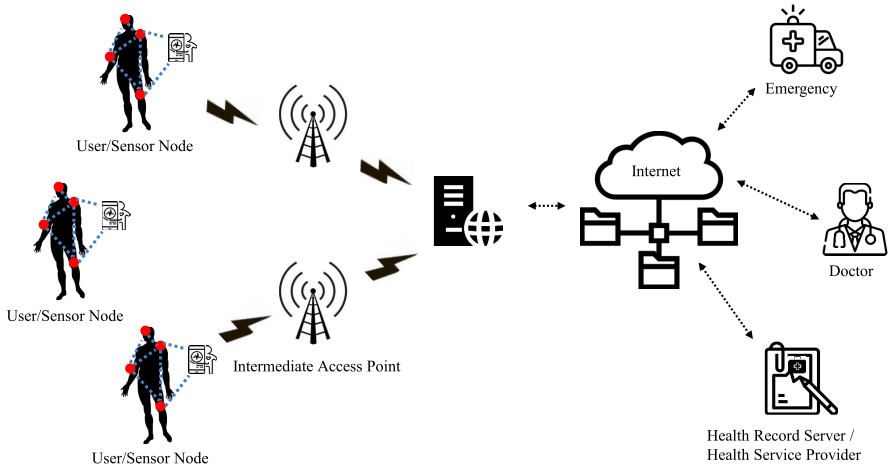

FIGURE 1. Typical wireless body area network system.

Figure 1 shows a general WBAN concept model, first designed by Zimmerman in 1996 [2]. Sensor devices located on the human body collect user healthcare data, such as heart rate, behavior, blood pressure, etc., and then transmit the biometric data to server. The server(s) subsequently communicate with doctors, healthcare service providers, emergency services, etc. to provide suitable medical services. Thus, it enables real-time patient condition monitoring and provides personalized healthcare services. However, these services are not secure against various potential attacks because an adversary can intercept, eavesdrop, reveal, delete, and/or modify data transmitted through public channels. Therefore, secure mutual authentication and key agreement schemes (MAKA) for medical IoT are important security issues to protect user health information while providing efficient healthcare services.

Many MAKA schemes have been presented over the preceding few decades to ensure user privacy. Lamport [3] first suggested an password based mutual authentication (MA) scheme in 1981, and several subsequent password based MA schemes have been proposed [4]–[6]. However, these designed protocols were vulnerable to various potential attacks, including privileged insider, impersonation and offline password guessing attacks, because they relied exclusively on the password. Many subsequent MAKA schemes have been designed to overcome these security weaknesses using smart cards [7]–[9] and/or biometrics [10]–[12]. However, these schemes stored user sensitive data in a server database, hence if the server stored data is revealed by an adversary, the whole system collapses.

Several authentication protocols have been designed for medical IoT to ensure user privacy [24]–[29]. However, these protocols were not secure against stolen verifier and/or leaking verification table attacks, not do they provide secure mutual authentication, untraceability, or anonymity.

Xu *et al.* [30] designed a lightweight MAKA scheme for medical IoT to prevent various attacks, including impersonation, replay and sensor node capture attacks. They also claimed their scheme provided sensor node anonymity and untraceability. However, we showed previously that the Xu *et al.*'s scheme does not provide anonymity and untraceability, nor does it prevent most attacks because their presented scheme stored user authentication parameters in a server database. Xu *et al.* also did not perform (mathematical) formal security analyses to prove their scheme security. For these reasons, we need provably secure and lightweight authentication scheme for medical IoT without verification table to protect user's medical data in which legal users and IoT things can securely authenticate and establish a session key. Therefore, we propose LAKS-NVT for medical IoT without requiring a verification table to overcome these observed security vulnerabilities.

B. RESEARCH CONTRIBUTIONS

The main contributions of this paper are as follows.

- We analyzed the Xu *et al.*'s scheme security vulnerabilities and demonstrate that it is not safe against impersonation, stolen SN, and leaking verification table attacks. We also prove that Xu *et al.*'s scheme does not ensure anonymity, secure mutual authentication, and untraceability.
- We propose LAKS-NVT for medical IoT without requiring a server verification table to resolve these security weaknesses. LAKS-NVT also prevents stolen SN, impersonation, and replay; and provides anonymity, secure mutual authentication, and untraceability. In addition, if the server verification table is leaked, LAKS-NVT is still secure because it does not store user's authentication parameters and sensitive data in the server's database.
- We perform (mathematical) formal security analysis using the Real-or-random (ROR) model [32] to prove session key security, and verified that LAKS-NVT provides secure MA using widely accepted Burrows-Abadi-Needham (BAN) logic [36].

- We also perform the formal security verification of the proposed LAKS-NVT using the widely-accepted "Automated Validation of Internet Security Protocols and Applications (AVISPA)" software tool [37] to show that it is secure.
- We analyze the performance of our scheme compared with other contemporary schemes, and then perform simulation analysis using the NS2 simulator.

C. THREAT MODEL

A server is generally considered as trustworthy node. However, an adversary can look up all parameters in the server's database, except the server master key, K_{ser}. An adversary can also eavesdrop, delete, replace, inject, and replay data transmitted in public channels. This case is called the Dolev-Yao (DY) threat model [31]. We assume that the sensor node (SN) is untrustworthy. After obtaining a SN, an adversary can extract and get data stored in the SN using the power analysis attacks [44], [45] and performs various potential attacks using this obtained data.

D. ORGANIZATION

The remainder of this paper is organized as follows. Section II discusses related works, and then Sections III reviews Xu *et al.*'s scheme and Section IV cryptanalyzes it. Section V details the proposed LAKS-NVT for medical IoT, and Sections VI and VIII analyze the proposed scheme security and practical demonstration using NS2 simulation study. Section X provides performance comparison with related schemes. Finally, Section XI concludes and summarizes this paper.

II. RELATED WORK

Many MAKA schemes as well as access control schemes for IoT and other related domains have been designed over the last few decades to guarantee user privacy and provide convenient services [13]–[28], [30], [38]–[43].

Liu *et al.* [24] suggested a "certificateless remote anonymous MAKA scheme" for WBANs, but this was subsequently shown to be vulnerable to "stolen verifier attacks" and could not provide scalability and forward secrecy [25]. Zhao [25] then designed an efficient authentication scheme for WBANs using an elliptic curve cryptosystem (ECC). Turkanovic *et al.* [38] proposed an MAKA scheme for IoT to provide "secure communication between user and sensor". However, Chang and Le [39] showed that scheme could not prevent various attacks, including impersonation, stolen smart card, sensor node spoofing, node capture, and stolen verifier attacks. They also suggested an "enhanced provably secure authentication scheme", considering flexibility and efficiency. Gope and Hwang [41] designed an anonymity-preserving MAKA for global mobility networks that guaranteed the communication security. However, Li *et al.* [40] showed that the Chang and Le scheme was not secure against trace and stolen smart card attack, and the Gope and Hwang scheme did not provide secure MA and an efficient password change phase. They suggested a robust and efficient MA protocol to overcome these security drawbacks. Li *et al.* [29] showed that the Gope and Hwang [41] scheme did not provide an "efficient verification mechanism" and perfect forward secrecy, and designed a robust biometrics-based MA scheme to resolve these security weaknesses. However, all the above schemes [24], [25], [29], [38]–[41] are somewhat inefficient and inapplicable for practical medical IoT environments, since they all use public key cryptosystems, which require high computational cost.

The several lightweight authentication and key agreement schemes [26]–[28], [30] have been proposed, considering computational costs. Ibrahim *et al.* [27] introduced a "secure and lightweight mutual authentication for WBANs" to provide anonymity and secure mutual authentication. Li *et al.* [28] designed an anonymous mutual authentication and key agreement scheme for WBANs that guaranteed anonymity and unlinkability for wearable sensors. Xu *et al.* [26] presented a lightweight mutual authentication and key agreement scheme for WBANs and claimed the scheme was secure against various attacks, including man-in-the-middle, spoofing, replay, and impersonation attacks. Xu *et al.* [30] subsequently introduced a "lightweight mutual authentication and key agreement scheme for medical IoT". However, they did not prove their scheme security using the (mathematical) formal security analysis. The above schemes [26]–[28], [30] are all vulnerable to leaking verification table attack because they store sensitive user data in a server database.

TABLE 1. Notation used in this paper.

Notation	Description
SA	System administrator
SN	Sensor node
IAP	Intermediate access point
S	Server node
K_{ser}	Master secret for server
ID_{SN}	Sensor node real identity
ID_{IAP}	IAP real identity
K_s	Session key between SN and S
P_{K_s}	Hash operation result for K_s
$h(\cdot)$	"Collision resistant cryptographic hash function"
$\oplus$	Bitwise XOR operation
$\|\|$	"Concatenation operation"

III. XU *et al.*'s SCHEME REVIEW

This section reviews the Xu *et al.*'s MAKA scheme for medical IoT [30]. The Xu *et al.*'s scheme comprises three phases: a) "initialization", b) "registration", and c) "mutual authentication and key agreement (MAKA)".

Table 1 shows the notation used in this paper.

A. INITIALIZATION PHASE

The SA establishes system parameters in this phase, first generating server master key, K_{ser}, and then storing it in the server memory.

Sensor node (SN)	**Server/system administrator (S/SA)**
	Select random number r, P_{K_s}, Assign identity ID_{SN} Compute $A_{SN} = r \oplus K_{ser}$, $B_{SN} = h(r\|\|K_{ser})$, $X = ID_{SN} \oplus h(r\|\|K_{ser})$ Store tuple (A_{SN}, X, P_{K_s}) in server memory $\{ID_{SN}, A_{SN}, B_{SN}, P_{K_s}\}$ $\xleftarrow{}$ (via secure channel)
Store $(ID_{SN}, A_{SN}, B_{SN}, P_{K_s})$ in SN memory	
Intermediate access point (IAP)	**Server/System Administrator (S/SA)**
	Select identity ID_{IAP} Store ID_{IAP} in server memory $\{ID_{IAP}\}$ $\xleftarrow{}$ (via secure channel)

FIGURE 2. Registration phase for the Xu *et al.*'s scheme.

B. REGISTRATION PHASE

This phase registers SNs and IAps, as shown in Fig. 2 with detailed steps as follows.

Step 1: S generates identity ID_{SN}, random number r, and P_{K_s} for each SN; and generates identity ID_{IAP} for each AP.

Step 2: SA computes $A_{SN} = r \oplus K_{ser}$, $B_{SN} = h(k||K_{ser})$, $X = ID_{SN} \oplus h(r||K_{ser})$, and then stores tuple $(ID_{SN}, A_{SN}, B_{SN}, P_{K_s})$ and (A_{SN}, X, P_{K_s}) in SN and S memory, respectively.

Step 3: Finally, SA stores ID_{IAP} in S memory.

C. MAKA PHASE

Fig. 3 shows that the SN and S authenticate each other and generate the current session key to access useful medical services, with detailed steps as below.

Step 1. SN generates a random number $n1$ and the current timestamp $t1$, and then sends the login request messages $\{A_{SN}, S1, S2, t1\}$ to the AP.

Step 2. After receiving messages $\{A_{SN}, S1, S2, t1\}$, IAP resends the data to S, including its own identity ID_{IAP}.

Step 3. Upon receiving messages $\{A_{SN}, S1, S2, t1, ID_{IAP}\}$ from IAP, S checks whether ID_{IAP} is in the database. If it does not exist, S discontinues the current session.

Step 4. S checks that $t_{new} - t1 < \Delta t$, where Δt and t_{new} are "maximum transmission delay" and reception time of messages, respectively. If not valid, S terminates the session; otherwise, S checks if A_{SN} exist in the database and retrieves tuple $\{A_{SN}, X, P_{K_s}\}$.

Step 5. S computes $r^* = A_{SN} \oplus K_{ser}, B^*_{SN} = h(r^*||K_{ser})$, $n1^* = S1 \oplus B^*_{SN}, ID^*_{SN} = X \oplus B^*_{SN}, S2^* = h(ID^*_{SN}||A_{SN}||S1||t1||n1^*)$; and then checks $S2^* \stackrel{?}{=} S2$.

Step 6. If it is valid, S chooses random number $n2$, timestamp $t2$, and unique number r^+; then computes $A^+_{SN} = r^+ \oplus K_{ser}, B^+_{SN} = h(r^+||K_{ser})$, $X^+ = ID^*_{SN} \oplus B^+_{SN}, S3 = n2 \oplus B^*_{SN}, y = h(ID^*_{SN}||n1^*||n2)$, $S4 = A^+_{SN} \oplus y \oplus n1^*, S5 = B^+_{SN} \oplus y, K_s = h(n1^*||n2||P_{K_s}$, and $P^+_{K_s} = h(K_s||n1^*||n2), S6 = h(S3||S4||S5||n2||ID^*_{SN}||P_{K_s}||t2)$.

Step 7. S replaces tuple $< A_{SN}, X, P_{K_s} >$ with $< A^+_{SN}, X^+, P^+_{K_s}, A_{SN}, X, P_{K_s} >$ in S memory; sends messages $\{S3, S4, S5, S6, t2, ID_{IAP}\}$ to IAP; and the stores session key K_s.

Step 8. Upon receiving messages $\{S3, S4, S5, S6, t2, ID_{IAP}\}$, IAP sends the data to SN except for its own identity, ID_{IAP}.

Step 9. After receiving messages $\{S3, S4, S5, S6, t2\}$ from IAP, SN checks whether $t_{new} - t2 < \Delta t$. If not valid, SN discontinues the current session; otherwise, SN computes $n2^* = S3 \oplus B_{SN}, S6^* = h(S3||S4||S5||n2^*||ID_{SN}||P_{K_s}||t2)$.

Step 10. SN checks whether $S6^* \stackrel{?}{=} S6$. If valid, SN computes $K_s = h(n1||n2^*||P_{K_s}), P^+_{K_s} = h(K_s||n1||n2^*)$, $y = h(ID^*_{SN}||n1||n2^*), A^+_{SN} = S4 \oplus y \oplus n1, B^+_{SN} = S5 \oplus y$; replaces $< A_{SN}, B_{SN}, P_{K_s} >$ with $< A^+_{SN}, B^+_{SN}, P^+_{K_s} >$ in SN memory; and stores session key K_s.

IV. CRYPTANALYSIS FOR THE XU *et al.*'s SCHEME

This section cryptanalyzes the Xu *et al.*'s scheme, and demonstrates the scheme does not prevent various attacks nor guarantee essential security requirements, such as "untraceability", "anonymity", and "secure mutual authentication".

A. STOLEN SERVER NODE ATTACK

Section I-C shows the DY threat model to evaluate the security of protocols in this paper. We suppose that an adversary U_A obtains SN for a legitimate user, intercepts transmitted messages in a public channel, and extracts values $\{ID_{SN}, A_{SN}, B_{SN},$ and $P_{K_s}\}$ using power analysis [44], [45]. Under Xu *et al.*'s scheme, authentication parameters $\{ID_{SN}, B_{SN}\}$ are stored as plaintext, and hence the scheme does not prevent stolen SN attack because U_A can perform various attacks using these security parameters.

B. IMPERSONATION ATTACK

Section IV-A shows how U_A can obtain SN parameters and messages transmitted via public channels. After obtaining these values, U_A generates a random nonce $n1_A$ and current timestamp $t1_A$, and computes $S1_A = B_{SN} \oplus n1_A$ and $S2_A = h(ID_{SN}||A_{SN}||S1||t1||n1_A)$. U_A can also retrieve $n2 = S3 \oplus B_{SN}$ and calculate $K_{sA} = h(n1_A||n2||P_{K_s})$. Therefore, Xu *et al.*'s scheme does not withstand impersonation attack since U_A can successfully compute login request, response messages and the session key.

Sensor node (SN)	Intermediate access point (IAP)	Server/system administrator (S/SA)
Choose $n1$ and timestamp $t1$ Compute $S1 = B_{SN} \oplus n1$, $S2 = h(ID_{SN}\|\|A_{SN}\|\|S1\|\|t1\|\|n1)$ $\{A_{SN}, S1, S2, t1\}$ (via open channel)		
	$\{A_{SN}, S1, S2, t1, ID_{IAP}\}$ (via open channel)	
		Check whether ID_{IAP} is in the database Check $t_{new} - t1 < \Delta t$ Check whether A_{SN} is in the database Retrieve the tuple $< A_{SN}, X, P_{K_s} >$ corresponding to A_{SN} Compute $r^* = A_{SN} \oplus K_{ser}, B^*_{SN} = h(r^*\|\|K_{ser})$ $n1^* = S1 \oplus B^*_{SN}, ID^*_{SN} = X \oplus B^*_{SN}$, $S2^* = h(ID^*_{SN}\|\|A_{SN}\|\|S1\|\|t1\|\|n1^*)$ Check $S_2^* \stackrel{?}{=} S2$ Generate random number, $n2$, timestamp $t2$, and unique random number r^+ Compute $A^+_{SN} = r^+ \oplus K_{ser}, B^+_{SN} = h(r^+\|\|K_{ser})$, $X^+ = ID^*_{SN} \oplus B^+_{SN}, S3 = n2 \oplus B^*_{SN}$, $y = h(ID^*_{SN}\|\|n1^*\|\|n2), S4 = A^+_{SN} \oplus y \oplus n1^*$, $S5 = B^+_{SN} \oplus y, K_s = h(n1^*\|\|n2\|\|P_{K_s})$, $P^+_{K_s} = h(K_s\|\|n1^*\|\|n2)$, $S6 = h(S3\|\|S4\|\|S5\|\|n2\|\|ID^*_{SN}\|\|P_{K_s}\|\|t2)$ Replace $< A_{SN}, X, P_{K_s} >$ with $< A^+_{SN}, X^+,$ $P^+_{K_s}, A_{SN}, X, P_{K_s} >$ Store session key K_s $\{S3, S4, S5, S6, t2, ID_{IAP}\}$ (via open channel)
	$\{S3, S4, S5, S6, t2\}$ (via open channel)	
Check whether $t2$ is valid Compute $n2^* = S3 \oplus B_{SN}$, $S6^* = h(S3\|\|S4\|\|S5\|\|n2^*\|\|ID_{SN}\|\|P_{K_s}\|\|t2)$ Check $S6^* \stackrel{?}{=} S6$ Computes $K_s = h(n1\|\|n2^*\|\|P_{K_s}), P^+_{K_s} = h(K_s\|\|n1\|\|n2^*)$, $y = h(ID^*_{SN}\|\|n1\|\|n2^*), A^+_{SN} = S4 \oplus y \oplus n1$, $B^+_{SN} = S5 \oplus y$ Replace $< A_{SN}, B_{SN}, P_{K_s} >$ with $< A^+_{SN}, B^+_{SN}, P^+_{K_s} >$ Store the session key K_s		

FIGURE 3. MAKA phase in Xu *et al.*'s scheme.

C. ANONYMITY AND UNTRACEABILITY

Section I-C shows how U_A can look up all parameters in the S database, except the server's master key, K_{ser}. After receiving login request $\{A_{SN}, S1, S2, t1, ID_{IAP}\}$, U_A checks whether A_{SN} exist in database. If it exist, U_A retrieves $A_{SN_{prev}}$ and traces a legal user by finding $A_{SN_{prev}}$, where $A_{SN_{prev}}$ is the parameter used in previous session. U_A can also obtain SN's real identity because ID_{SN} is stored in SN memory as plaintext. Therefore, Xu *et al.*'s scheme does not ensure untraceability and anonymity.

D. MUTUAL AUTHENTICATION (MA)

From Section IV-B, U_A can compute login request messages $\{A_{SN}, S1, S2, t1\}$ and verify response messages $\{S3, S4, S5, S6, t2\}$ successfully. Furthermore, U_A can obtain the session key, K_S. Thus, Xu *et al.*'s scheme does not ensure secure MA.

E. LEAKING VERIFICATION TABLE ATTACK

Section I-C shows how U_A can obtain SN authentication parameters and data stored in the server database except the server's master key, K_{ser}. If U_A obtains the partial user dataset, e.g. $\{ID_{SN}, A_{SN}\}$, $\{ID_{SN}, B_{SN}\}$, or $\{B_{SN}, X\}$, U_A can successfully perform an impersonation attack. U_A also breaks anonymity, untraceability, and secure mutual authentication between SN and S. Therefore, it is important that essential parameters be managed directly by the user.

User/sensor node (U/SN)	Server/system administrator (S/SA)
Choose ID_U, PW_U, random number r, k Compute $PID = h(ID_U \|\| PW_U) \oplus r$, $\{PID, r\}$ → (via secure channel)	
	Compute $S = h(PID \|\| K_{ser})$, $A_{SN} = r \oplus S$, $B_{SN} = h(r \|\| PID \|\| K_{ser})$ $\{S, A_{SN}, B_{SN}\}$ ← (via secure channel)
Compute $SPW = h(PW_U \|\| k)$, $S' = S \oplus h(ID_U \|\| PW_U)$, $V = h(ID_U \|\| PW_U \|\| k)$, $C = h(ID_U \|\| PW_U) \oplus k$, $CPID = PID \oplus SPW$, $CB_{SN} = B_{SN} \oplus h(ID_U \|\| k)$, $CA_{SN} = A_{SN} \oplus h(B_{SN} \|\| k)$ Store $CPID, V, C, CB_{SN}$, and CA_{SN} in SN memory	
Intermediate access point (IAP)	**Server/system administrator (S/SA)**
	Select identity ID_{IAP} Store ID_{IAP} in the memory $\{ID_{IAP}\}$ ← (via secure channel)

FIGURE 4. Registration phase for our scheme.

V. THE PROPOSED SCHEME

This section proposes LAKS-NVT for medical IoT without requiring a server verification table to overcome security flaws in the Xu *et al.*'s scheme as shown in in Section IV. The designed scheme also includes three phases: a) ''initialization'', b) ''registration'', and c) ''mutual authentication and key agreement (MAKA)''.

A. INITIALIZATION PHASE

The SA establishes the system parameters, generates a master key, K_{ser}, for the server, and stores it in server memory.

B. REGISTRATION PHASE

When user U wants to use medical services from S, U must first register their identity with S. The *IAP* provides a connecting node between SN and S. This phase is shown in Fig. 4 with detailed steps as follows.

1) **User registration phase**

Step 1: U picks identity ID_U, password PW_U, random number r, k; computes $PID = h(ID_U||PW_U) \oplus r$; and sends $\{PID, r\}$ to S.

Step 2: After receiving $\{PID, r\}$, S computes $S = h(PID||K_{ser}$, $A_{SN} = r \oplus S$, and $B_{SN} = h(r||PID||K_{ser})$; and then sends $\{S, A_{SN}, B_{SN}\}$ to U

Step 3: U computes $SPW = h(PW_U||k)$, $S' = S \oplus h(ID_U||PW_U)$, $V = h(ID_U||PW_U||k)$, $C = h(ID_U||PW_U) \oplus k$, $CPID = PID \oplus SPW$, $CB_{SN} = B_{SN} \oplus h(ID_U||k)$, and $CA_{SN} = A_{SN} \oplus h(B_{SN}||k)$; and then stores $CPID, V, C, CB_{SN}, CA_{SN}$ in SN memory.

2) **Intermediate access point registration phase** Server S picks ID_{IAP} and sends it to *IAP* through a secure channel. Then, S and *IAP* store ID_{IAP} in the database, respectively.

C. MAKA PHASE

User U sends MAKA request messages to access medical services, as shown in Fig. 5 with detailed steps are as follows.

Step 1. SN generates random number $n1$ and current timestamp $t1$, and then sends login request messages $\{A_{SN}, S1, S2, t1\}$ to the AP.

Step 2. After receiving messages $\{A_{SN}, S1, S2, t1\}$, *IAP* resends the data to S, including its own identity ID_{IAP}.

Step 3. upon receiving messages $\{A_{SN}, S1, S2, t1, ID_{IAP}\}$ from IAP, S checks whether ID_{IAP} is in the database. If it does not exist, S terminates the current session.

Step 4. S checks $t_{new} - t1 < \Delta t$, where Δt and t_{new} are maximum transmission delay and message reception time, respectively. If not valid, S terminates the session; otherwise, S checks if A_{SN} exists in the database and retrieves tuple $\{A_{SN}, X, P_{K_s}\}$.

Step 5. S computes $r^* = A_{SN} \oplus K_{ser}$, $B^*_{SN} = h(r^*||K_{ser})$, $n1^* = S1 \oplus B^*_{SN}$, $ID^*_{SN} = X \oplus B^*_{SN}$, $S2^* = h(ID^*_{SN}||A_{SN}||S1||t1||n1^*)$, and then checks $S2^* \stackrel{?}{=} S2$.

Step 6. If valid, S chooses random number $n2$, timestamp $t2$ and unique number r^+; and then computes $A^+_{SN} = r^+ \oplus K_{ser}$, $B^+_{SN} = h(r^+||K_{ser})$, $X^+ = ID^*_{SN} \oplus B^+_{SN}$, $S3 = n2 \oplus B^*_{SN}$, $y = h(ID^*_{SN}||n1^*||n2)$, $S4 = A^+_{SN} \oplus y \oplus n1^*$, $S5 = B^+_{SN} \oplus y$, $K_s = h(n1^*||n2||P_{K_s})$, and $P^+_{K_s} = h(K_s||n1^*||n2)$, $S6 = h(S3||S4||S5||n2||ID^*_{SN}||P_{K_s}||t2)$.

Step 7. S replaces tuple $< A_{SN}, X, P_{K_s} >$ with $< A^+_{SN}, X^+, P^+_{K_s}, A_{SN}, X, P_{K_s} >$ in S memory; sends messages $\{S3, S4, S5, S6, t2, ID_{IAP}\}$ to IAP; and stores session key K_s.

Step 8. Upon receiving messages $\{S3, S4, S5, S6, t2, ID_{IAP}\}$, *IAP* sends the data to SN except for its own identity ID_{IAP}.

Step 9. After receiving messages $\{S3, S4, S5, S6, t2\}$ from IAP, SN checks $t_{new} - t2 < \Delta t$. If not valid, SN terminates the current session; otherwise, it computes $n2^* = S3 \oplus B_{SN}$, $S6^* = h(S3||S4||S5||n2^*||ID_{SN}||P_{K_s}||t2)$.

Step 10. SN checks $S6^* \stackrel{?}{=} S6$. If valid, SN computes $K_s = h(n1||n2^*||P_{K_s})$, $P^+_{K_s} = h(K_s||n1||n2^*)$, $y = h(ID^*_{SN}||n1||n2^*)$, $A^+_{SN} = S4 \oplus y \oplus n1$, $B^+_{SN} = S5 \oplus y$; replaces $< A_{SN}, B_{SN}, P_{K_s} >$ with $< A^+_{SN}, B^+_{SN}, P^+_{K_s} >$ in SN memory; and stores session key K_s.

User (U)/Sensor node (SN)	Intermediate access point (IAP)	Server/system administrator (S/SA)
User U inputs ID_U, PW_U Generate $n1$ and timestamp $t1$ Compute $k = C \oplus h(ID_U \| PW_U)$, $SPW = h(PW_U \| k)$, $h(ID_U \| K), h(B_{SN} \| k)$ Retrieve $PID = CPID \oplus SPW$, $B_{SN} = CB_{SN} \oplus h(ID_U \| k)$, $A_{SN} = CA_{SN} \oplus h(B_{SN} \| k)$ and compute $S1 = B_{SN} \oplus n1, V' = h(ID_U \| PW_U \| k)$, $S2 = h(ID_{SN} \| A_{SN} \| S1 \| t1 \| n1)$ Check $V' \stackrel{?}{=} V$ $\{PID, A_{SN}, S1, S2, t1\}$ $\longrightarrow$ (via open channel)		
	$\{PID, A_{SN}, S1, S2, t1, ID_{IAP}\}$ $\longrightarrow$ (via open channel)	
		Check $t_{new} - t1 < \Delta t$ Compute $S^* = h(PID \| K_{ser})$ Retrieve $r^* = A_{SN} \oplus S^*$ Compute $B^*_{SN} = h(r \| PID \| K_{ser})$, $n1^* = S1 \oplus B^*_{SN}$ $S2^* = h(PID \| A_{SN} \| S1 \| t1 \| n1^*)$ Check $S^*_2 \stackrel{?}{=} S2$ Generate random number $n2$, timestamp $t2$, unique random number $r^+ = h(n1^* \| n2)$ Compute $S3 = n2 \oplus B^*_{SN}, y = h(PID \| n1 \| n2)$, $PID^+ = r^+ \oplus r \oplus PID, S^+ = h(PID^+ \| K_{ser})$ $B^+_{SN} = h(r^+ \| PID^+ \| K_{ser}), S4 = y \oplus S^+$, $S5 = B^+_{SN} \oplus y$, $S6 = h(S3 \| S4 \| S5 \| n2 \| ID^*_{SN} \| P_{K_s} \| t2)$ session key $K_s = h(PID^+ \| n1^* \| n2 \| r^*)$ $\{S3, S4, S5, S6, t2, ID_{IAP}\}$ $\longrightarrow$ (via open channel)
	$\{S3, S4, S5, S6, t2\}$ $\longrightarrow$ (via open channel)	
Check whether $t2$ is valid Compute $n2^* = S3 \oplus B_{SN}, r^{**} = h(n1 \| n2)$, $y^* = h(PID \| n1 \| n2^*), PID^{**} = r^{**} \oplus r \oplus PID$ Retrieve $S^{**} = S4 \oplus y^*, B^{**}_{SN} = S5 \oplus y^*$ Compute $S6^* = h(S3 \| S4 \| S5 \| n2^* \| PID^{**} \| r^{**} \| t2)$ Check $S6^* \stackrel{?}{=} S6$ Compute session key $K_s = h(PID^{**} \| n1 \| n2^* \| r)$, $A^+_{SN} = r \oplus r^{**} \oplus S^{**}, S'_{new} = S^{**} \oplus h(ID_U \| PW_U)$ $CA^+_{SN} = A^+_{SN} \oplus B^{**}_{SN}, CB^+_{SN} = B^{**}_{SN} \oplus h(ID_U \| k)$, $CPID^+ = PID^{**} \oplus SPW$ Replace $< CA_{SN}, CB_{SN}, CPID, S >$ with $< CA^+_{SN}, CB^+_{SN}, CPID^+, S'_{new} >$		

FIGURE 5. MAKA phase for our scheme.

VI. SECURITY ANALYSIS

This section discusses a mathematical security analysis using the ROR model [32] and BAN logic [36], and informal (non-mathematical) security analysis to prove LAKS-NVT is secure against potential attacks, including stolen SN, impersonation, and replay attacks. We also demonstrate the scheme achieves secure MA, anonymity, untraceability, and session key security.

Wang *et al.* [47] observed while analyzing several existing authentication protocols that the broadly-used formal security methods, such as "random oracle model" and "BAN logic" can not capture some structural mistakes. As a result, they pointed out that guaranteeing soundness of authenticated key agreement protocols still remains an open issue. Due to this, we require formal and informal security analysis along with the formal security verification using automated

TABLE 2. Queries and descriptions.

Query	Role
$Execute(\mathcal{P}_{SN}^{t_1}, \mathcal{P}_{S}^{t_2})$	It is modeled as an eavesdropping attack: $\mathcal{A}$ can eavesdrop messages transmitted between entities SN and S through public channels.
$CorruptSN(\mathcal{P}_{EV_i}^{t})$	It is modeled as an active attack: $\mathcal{A}$ can obtain all sensitive data stored in the sensor node.
$Reveal(\mathcal{P}^t)$	$\mathcal{A}$ obtain session key K_S established between $\mathcal{P}^t$ and its partner using this query.
$Send(\mathcal{P}^t, Msg)$	It is modeled as an active attack: $\mathcal{A}$ can send message Msg to $\mathcal{P}^t$, and then receives response messages from $\mathcal{P}^t$".
$Test(\mathcal{P}^t)$	$\mathcal{A}$ requests $\mathcal{P}^t$ for session key K_S, and receives probabilistic output for an unbiased coin c. If K_S is fresh ($c = 1$), $\mathcal{P}^t$ receives K_S; otherwise, $\mathcal{P}^t$ receives a random number ($c = 0$) or null value ($\perp$).

validation tools so that the proposed scheme (LAKS-NVT) will be secure against possible potential attacks with high probability.

A. FORMAL SECURITY ANALYSIS USING THE ROR MODEL

In this section, we first briefly discuss the ROR model [32]. After that the formal (mathematical) security analysis under the ROR model is presented to prove the session key security for the proposed scheme (LAKS-NVT) in Theorem 1.

Based on the ROR model, a malicious adversary $\mathcal{A}$ communicates with the t-th of participant instance, $\mathcal{P}^t$. Following the proposed scheme, we define SN or S as $\mathcal{P}^t$ and let $\mathcal{P}_{SN}^{t_1}$ and $\mathcal{P}_{S}^{t_2}$ be the t-th SN and S instances, respectively. The ROR model uses *Execute*, *Reveal*, *CorruptSN*, *Send* and *Test* queries to simulate an actual attack, as shown in Table 2. We use a collision resistant one-way hash function *Hash*, i.e., $h(\cdot)$), as a random oracle — a deterministic function that outputs a fixed length string.

Wang *et al.* [46] demonstrated that user chosen passwords are extremely non-uniformly distributed using Zipf's law. Password dictionary size is also limited since users do not utilize the whole dictionary extent for passwords [46]. Zipf's law is widely applied in formal (mathematical) security analysis to prove session key security for cryptographic protocols. We now prove that the proposed LAKS-NVT achieves session key security.

1) SESSION KEY SECURITY

Based on the ROR model, $\mathcal{A}$ tries to distinguish between a session key and a random number using the several *Test* queries. After obtaining the result of *Test* query, $\mathcal{A}$ checks whether the guessed coin c' is consistent against the coin c of a real session key. If $c' = c$, $\mathcal{A}$ wins the game and its probability is *Succ*.

Theorem 1: Assume that $Adv_{\mathcal{A}}^{AKM}$ is the advantage of an adversary $\mathcal{A}$ running in polynomial time to break *session key security* for the proposed authenticated key management (AKM) scheme. Then,

$$Adv_{\mathcal{A}}^{AKM} \leq \frac{q_h^2}{|Hash|} + 2\max\left\{C' \cdot q_s^{s'}\right\},$$

where q_h, q_s, and $|Hash|$ are the number of *Hash* and *Send* queries, and the hash function $h(\cdot)$ output string length, respectively; and C' and s' are Zipf's parameters [46].

Proof: We define a sequence of four games, G_j, where $j = 0, 1, 2, 3$, with probability $Succ_{\mathcal{A}}^{G_j}$ that $\mathcal{A}$ correctly guesses random bit c in G_j. $\mathcal{A}$'s advantage of winning G_j is defined as $Adv_{\mathcal{A},G_j}^{AKE} = Pr[Succ_{\mathcal{A}}^{G_j}]$. The proof follows from [5], [33]–[35] as shown below.

- **Game** G_0 is simulated as the actual attack by $\mathcal{A}$ against the proposed protocol. Since bit c was chosen randomly at the beginning of G_0,

$$Adv_{\mathcal{A}}^{AKM} = |2.Adv_{\mathcal{A},G_0}^{AKM} - 1|. \quad (1)$$

- **Game** G_1 is modeled as an eavesdropping attack where $\mathcal{A}$ can eavesdrop all public messages, including $\{PID, A_{SN}, S1, S2, t1, ID_{IAP}\}$, and $\{S3, S4, S5, S6, t2\}$. Under G_1, $\mathcal{A}$ uses the *Execute* query, and performs *Reveal* and *Test* queries to check whether the derived session key K_s between SN and S is correct or random. Define $K_s = h(PID^{new}||n1||n2||r)$ because $\mathcal{A}$ needs short-term secret values ($n1$, $n2$, r and PID) and also master secret value ($K_s er$) to compute K_s correctly. Eavesdropped messages do not help $\mathcal{A}$ increase winning probability for G_1. Therefore, G_0 and G_1 are indistinguishable, and

$$Adv_{\mathcal{A},G_1}^{AKM} = Adv_{\mathcal{A},G_0}^{AKM}. \quad (2)$$

- **Game** G_2 was simulated as an active attack by *Hash* query. $\mathcal{A}$ wants to find message digest collision to deceive a participant using several *Hash* queries. However, all transmitted messages $\{PID, A_{SN}, S1, S2, t1, ID_{IAP}\}$ and $\{S3, S4, S5, S6, t2\}$ are protected by short-term secret, master secret, and timestamp. Therefore, G_1 and G_2 are indistinguishable because the collision probability is negligible when $\mathcal{A}$ sends several $Send(\mathcal{P}^t, Msg)$ queries. Thus, from the birthday paradox,

$$|Adv_{\mathcal{A},G_1}^{AKM} - Adv_{\mathcal{A},G_2}^{AKM}| \leq \frac{q_h^2}{2|Hash|}. \quad (3)$$

- **Game** G_3 was modeled as an active attack. $\mathcal{A}$ obtains credentials $\{CPID, V, C, CB_{SN}, CA_{SN}\}$ from SN's memory using the $CorruptSN(\mathcal{P}_{EV_i}^{t})$ query, where $PID = h(ID_U||PW_U) \oplus r$, $V = h(ID_U||PW_U||k)$, $C = h(ID_U||PW_U) \oplus k$, $SPW = h(PW_U||k)$, $B_{SN} = h(r||PID||K_{ser})$, $A_{SN} = r \oplus S$, and $S = h(PID||K_{ser})$. $\mathcal{A}$ must know the user's real identity ID_U, password PW_U, and short-term secrets r to retrieve secret parameters B_{SN} and A_{SN}. However, since $\mathcal{A}$ does not know B_{SN}, A_{SN}, K_{ser}, or r, they cannot correctly guess PW_i for SN using *Send* query. Therefore, games G_2 and G_3 are indistinguishable, from Zipf's law on passwords [46],

$$|Adv_{\mathcal{A},G_2}^{AKM} - Adv_{\mathcal{A},G_3}^{AKM}| \leq \max\left\{C' \cdot q_s^{s'}\right\}. \quad (4)$$

TABLE 3. BAN logic notations used here.

Notation	Description
$P\mid\equiv X$	P **believes** statement X
#X	Statement X is **fresh**
$P \lhd X$	P **see** statement X
$P\mid\sim X$	P once **said** X
$P \Rightarrow X$	P **controls** statement X
$< X >_Y$	Statement X is **combined** with secret statement Y
$\{X\}_K$	Statement X is **masked** by secret statement (key) K
$P \overset{K}{\leftrightarrow} Q$	P and Q share secret statement (key) K to communicate with each other

After all the games are executed, $\mathcal{A}$ tries to guess c to win the game using *Test* query. Therefore,

$$Adv^{AKM}_{\mathcal{A},G_3} = \frac{1}{2}. \tag{5}$$

Combining (1), (2), and (5),

$$\begin{aligned}\frac{1}{2}.Adv^{AKM}_{\mathcal{A}} &= |Adv^{AKM}_{\mathcal{A},G_0} - \frac{1}{2}| \\ &= |Adv^{AKM}_{\mathcal{A},G_1} - \frac{1}{2}| \\ &= |Adv^{AKM}_{\mathcal{A},G_1} - Adv^{AKM}_{\mathcal{A},G_3}|. \end{aligned} \tag{6}$$

From the triangular inequality with (4), (5), and (6),

$$\begin{aligned}\frac{1}{2}.Adv^{AKM}_{\mathcal{A}} &= |Adv^{AKM}_{\mathcal{A},G_1} - Adv^{AKM}_{\mathcal{A},G_3}| \\ &\leq |Adv^{AKM}_{\mathcal{A},G_1} - Adv^{AKM}_{\mathcal{A},G_2}| \\ &\quad + |Adv^{AKM}_{\mathcal{A},G_2} - Adv^{AKM}_{\mathcal{A},G_3}| \\ &\leq \frac{q_h^2}{2|Hash|} + \max\left\{C' \cdot q_s^{s'}\right\}. \end{aligned} \tag{7}$$

Finally, multiplying both sides of (7) by 2,

$$Adv^{AKM}_{\mathcal{A}} \leq \frac{q_h^2}{|Hash|} + 2\max\left\{C' \cdot q_s^{s'}\right\}.$$

B. SECURITY ANALYSIS USING BAN LOGIC

This section demonstrates that LAKS-NVT achieves secure mutual authentication using the BAN logic [36]. We first present the BAN logic postulates, and then define the security goals, assumptions, and idealized forms. Finally, we perform BAN logic proof to confirm secure MA for LAKS-NVT.

It is worth noticing that by the BAN logic proof, we only provide the mutual authentication proof of LAKS-NV among a user (U)/sensor node (SN) and the server/system administrator (S/SA) during the mutual authentication and key agreement phase. Table 3 presents BAN logic notations used in this proof.

1) BAN LOGIC POSTULATES

The BAN logic postulates are as follows.

- Message meaning rule (*MMR*):

$$\frac{P \mid\equiv P \overset{K}{\leftrightarrow} Q, \quad P \lhd \{X\}_K}{P \mid\equiv Q \mid\sim X}$$

- Nonce verification rule (*NVR*):

$$\frac{P \mid\equiv \#(X), \quad P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$$

- Jurisdiction rule (*JR*):

$$\frac{P \mid\equiv Q \mid\Longrightarrow X, \quad P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$$

- Freshness rule (*FR*):

$$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X, Y)}$$

- Belief rule (*BR*):

$$\frac{P \mid\equiv (X, Y)}{P \mid\equiv X.}$$

2) GOALS AND ASSUMPTIONS

We define goals ($Goal_1$–$Goal_4$) and assumptions (A_1–A_6) as follows to verify the proposed protocol security.

$Goal_1$: $SN \mid\equiv S \mid\equiv (SN \overset{K_s}{\longleftrightarrow} S)$
$Goal_2$: $SN \mid\equiv (SN \overset{K_s}{\longleftrightarrow} S)$
$Goal_3$: $S \mid\equiv SN \mid\equiv (SN \overset{K_s}{\longleftrightarrow} S)$
$Goal_4$: $S \mid\equiv (SN \overset{K_s}{\longleftrightarrow} S)$

A_1: $S \mid\equiv (SN \overset{r}{\longleftrightarrow} S)$
A_2: $S \mid\equiv \#(n1)$
A_3: $SN \mid\equiv (SN \overset{r^+}{\longleftrightarrow} S)$
A_4: $SN \mid\equiv \#(n2)$
A_5: $S \mid\equiv SN \mid\Rightarrow (K_s)$
A_6: $SN \mid\equiv S \mid\Rightarrow (K_s)$

3) IDEALIZED FORMS

The idealized forms are below.

Msg_1: $SN \rightarrow S$: $(PID, K_{ser}, n1, t1)_r$
Msg_2: $S \rightarrow SN$: $(n2, n1, PID^+, K_{ser}, t2)_{r^+}$

4) BAN LOGIC PROOF

We perform BAN logic analysis to verify that LAKS-NVT guarantees secure MA.

Step 1. From Msg_1,

$$S_1 : S \lhd (PID, K_{ser}, n1, t1)_r.$$

Step 2. From the *MMR* S_1 and A_1,

$$S_2 : S \mid\equiv SN \mid\sim (PID, K_{ser}, n1, t1)_r.$$

Step 3. From the *FR* with A_2,

$$S_3 : S \mid\equiv \#(PID, K_{ser}, n1, t1)_r.$$

Step 4. From the *NVR* with S_2 and S_3,

$$S_4 : S \mid\equiv SN \mid\equiv (PID, K_{ser}, n1, t1)_r.$$

Step 5. From Msg_2,

$$S_5 : SN \lhd (n2, n1, PID^+, K_{ser}, t2)_{r^+}.$$

Step 6. From the *MMR* with S_5 and A_3,

$$S_6 : SN \mid\equiv S \mid\sim (n2, n1, PID^+, K_{ser}, t2)_{r^+}.$$

tep 7. From the *FR* with A_4,

$$S_7 : SN \mid\equiv \#(n2, n1, PID^+, K_{ser}, t2)_{r^+}.$$

Step 8. From the *NVR* with S_6 and S_7,

$$S_4 : SN \mid\equiv S \mid\equiv (n2, n1, PID^+, K_{ser}, t2)_{r^+}.$$

Step 9. From S_4 and S_8, S and SN can compute session key K_s because they trust each other by BAN logic postulates,

$$S_9 : SN \mid\equiv S \mid\equiv (SN \overset{K_s}{\longleftrightarrow} S) \qquad Goal1$$

and

$$S_{10} : S \mid\equiv SN \mid\equiv (SN \overset{K_s}{\longleftrightarrow} S) \qquad Goal3.$$

Step 10. From the *JR* with S_9 and A_5,

$$S_{11} : S \mid\equiv (SN \overset{K_s}{\longleftrightarrow} S) \qquad Goal4.$$

Step 11. From the *JR* with S_{10} and A_6,

$$S_{11} : SN \mid\equiv (SN \overset{K_s}{\longleftrightarrow} S) \qquad Goal2.$$

Therefore, ($Goal_1$–$Goal_4$) prove that SN and S can trust each other and LAKS-NVT achieves secure MA.

C. INFORMAL SECURITY ANALYSIS

We analyzed LAKS-NVT for various potential attacks.

1) STOLEN SN ATTACK

Assume that attacker U_A obtains the SN for some legitimate user, intercepts public messages, and extracts SN values $\{CPID, V, C, CB_{SN}, andCA_{SN}\}$ using power analysis [44], [45]. However, although U_A obtains these parameters, they cannot obtain user credentials, such as short-term secrets r or authentication parameters $A_{SN}andB_{SN}$ without knowing the user's ID_U and PW_U. Thus, LAKS-NVT is secure against stolen SN attack.

2) IMPERSONATION ATTACK

If U_A wants to impersonate legitimate user U_i, they must correctly compute login request messages $\{PID, A_{SN}, S1, S2, t1\}$, where, $PID = CPID \oplus SPW$, $SPW = h(PW_U||k)$, $A_{SN} = CA_{SN} \oplus h(B_{SN}||k)$, $B_{SN} = CB_{SN} \oplus h(ID_U||k)$, $S1 = B_{SN} \oplus n1$, and $S2 = h(ID_{SN}||A_{SN}||S1||t1||n1)$. Since all login parameters are protected by hash function and secret parameters $\{k, r, ID_U andPW_U\}$, $U_A\}$ cannot compute the login request messages without knowing ID_U and PW_U. Therefore, LAKS-NVT prevents impersonation attack.

3) REPLAY ATTACK

LAKS-NVT is secure against replay attack since it uses a timestamp for all transmitted messages. If U_A resends some previous login request messages $\{PID, A_{SN}, S1, S2, t1\}$ to impersonate a real user, they must successfully compute $S2 = h(ID_{SN}||A_{SN}||S1||t1||n1)$ and $S6 = h(S3||S4||S5||n2||ID_{SN}||P_{K_s}||t2)$. However, U_A requires authentication parameters $A_{SN}, B_{SN}, n1, n2$ and ID_{SN} to compute these values. Thus, LAKS-NVT resists replay attack.

4) ANONYMITY AND UNTRACEABILITY

Suppose U_A obtains $CPID, V, C, CB_{SN}, CA_{SN}$ from SN memory and intercepts all previous messages to try to obtain ID_U. U_A cannot obtain ID_U since the user only employs the pseudo-identity PID to authenticate with the server. Since all transmitted messages ($\{PID, A_{SN}, S1, S2, t1\}$, $\{S3, S4, S5, S6, t2\}$) change every session, U_A is also unable to successfully perform trace attack. Therefore, LAKS-NVT achieves anonymity and untraceability.

5) SECURE MUTUAL AUTHENTICATION (MA)

From Section VI-C2, U_A cannot successfully compute valid login request and response messages. SN and S also check $S6^* \overset{?}{=} S6$ and $S2^* \overset{?}{=} S2$ are valid during the authentication and key agreement phase. Therefore, LAKS-NVT guarantees secure MA.

6) LEAKING VERIFICATION TABLE ATTACK

If U_A obtains SN information $CPID, V, C, CB_{SN}, CA_{SN}$ and accesses the server database, they cannot obtain user sensitive data because user information is not stored in the server database. All authentication parameters are changed every session and user manages it on himself/herself. Therefore, although the server database is compromised, LAKS-NVT remains secure against potential attacks.

VII. FORMAL SECURITY VERIFICATION USING AVISPA

This section provides the formal security verification of the proposed scheme (LAKS-NVT) using one of widely-accepted automated validation software tools, known as "Automated Validation of Internet Security Protocols and Applications (AVISPA)" tool [48].

A. AVISPA OVERVIEW

AVISPA is a formal security verification tool that proves whether a protocol is secure against "replay" and "man-in-the-middle" attacks. The "High-Level Protocol specification Language" (HLPSL) is used to implement a protocol. There are four backends related to AVISPA tool, which include: a) "On-the-fly Model-Checker (OFMC)", b) "Constraint Logic based Attack Searcher (CL-AtSe)", c) "SAT-based Model-Checker (SATMC)", and d) "Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP)". The HLPSL code is converted to the

"Intermediate Format (IF)", which is done with the help of HLPSL2IF translator. The IF is then provided as an input to one of the four backends, which in turn produce the "Output Format (OF)". The OF consists of the following sections [48]:

- SUMMARY: It indicates "whether the tested protocol is safe, unsafe, or whether the analysis is inconclusive".
- DETAILS: It tells a "detailed explanation of why the tested protocol is concluded as safe, or under what conditions the test application or protocol is exploitable using an attack, or why the analysis is inconclusive".
- PROTOCOL: It defines the "HLPSL specification of the target protocol in intermediate form".
- GOAL: It indicates "the goal of the analysis which is being performed by AVISPA using HLPSL specification".
- BACKEND: It is "the name of the back-end that is used for the analysis, that is, one of OFMC, CL-AtSe, SATMC and TA4SP".
- Final section tells about the "trace of a possible vulnerability to the target protocol, if any, along with some useful statistics and relevant comments".

The basic types supported by HLPSL are as follows:

- *agent:* It indicates a "principal name". The intruder is always denoted by *i* and considered as a legitimate entity in the specification of the protocol.
- *symmetric_key:* The keys that are relevant in the context of a "symmetric-key cryptosystem" are declared in this category.
- *text:* It usually represents a random nonce. It is also used sometimes to declared as the messages.
- *nat:* Under this category, the natural numbers are denoted in non-message contexts.
- *const:* Under this type, the constants in the protocol specification are declared.
- *hash_func:* This type indicates the "one-way cryptographic hash functions", which are treated as non-invertible functions.

Given a "plaintext message, say m" and an "encryption key k", the symmetric/public-key encryption of m using the key k is defined by $\{m\}_k$. The concatenation of In HLPSL syntax, two messages/strings X and Y are concatenated by $X \cdot Y$ using the "$\cdot$" operator, which follows the "associative rule".

B. HPLSL IMPLEMENTATION

The proposed scheme (LAKS-NVT) has been implemented using the HPLSL. In this implementation, we have three basic roles for a user/sensor node, a server/system administrator and the intermediate access point, which are shown in Figures 6, 7 and 8. Apart from these three basic roles, we have defined two mandatory roles for the "session" and "goal and environment" which are defined in Figure 9.

Consider the basic role of a user/sensor node in Figure 6 where user registration and authentication phases are

```
role user(USN, SSA, IAP: agent, H: hash_func, Snd, Rcv: channel(dy))
%%% USN: user/sensor node, SSA: server/system administrator;
%%% IAP: Intermediate access point
% Player: User/Sensor Node (USN)
played_by USN
def=
local State: nat,
    SKus: symmetric_key,
    IDu, PWu, R, K, Kser, PID, Pks, N1, T1, N2, T2: text,
 Asn1, S1, R1, S2, IDsn: text
const sp1, sp2, sp3, usn_ssa_t1, usn_ssa_n1,
    ssa_usn_t2, ssa_usn_n2: protocol_id
init  State := 0

transition
%%% User registration phase
1. State = 0 ∧ Rcv(start) =|>
State' := 1 ∧ K' := new() ∧ R' := new()
        ∧ PID' := xor(H(IDu.PWu), R')
%% Send registration request to SSA via secure channel
        ∧ Snd({PID'.R'}_SKus)
        ∧ secret({IDu,PWu,K'}, sp1, {USN})
        ∧ secret({R',Pks}, sp2, {USN,SSA})
%%% Receive registration response from SSA via secure channel
2. State = 1 ∧ Rcv({H(xor(H(IDu.PWu), R').Kser).
                xor(R', H(xor(H(IDu.PWu), R').Kser)).
                H(R'.xor(H(IDu.PWu), R').Kser)}_SKus)  =|>
State' := 2  ∧ secret({Kser}, sp3, {SSA})
%%% Authentication phase
        ∧ N1' := new() ∧ T1' := new()
        ∧ Asn1' := xor(R', H(xor(H(IDu.PWu), R').Kser))
        ∧ S1' := xor(H(R'.xor(H(IDu.PWu), R').Kser), N1')
        ∧ S2' := H(IDsn.Asn1'.S1'.T1'.N1')
%%% Send message to IAP/SSA via public channel
        ∧ Snd(PID'.Asn1'.S1'.S2'.T1')
%% U has freshly generated the values t1 and n1 for IAP/SSA
        ∧ witness(USN, SSA, usn_ssa_t1, T1')
        ∧ witness(USN, SSA, usn_ssa_n1, N1')
%%% Receive message from IAP/SSA via public channel
3. State = 2 ∧ Rcv(xor(N2', H(R'.xor(H(IDu.PWu), R1').Kser)).
              xor(H(xor(H(IDu.PWu), R').N1'.N2'), xor(H(IDu.PWu), R1')).
              xor(H(R1'.xor(H(IDu.PWu), R1').Kser), H(xor(H(IDu.PWu), R').
              N1'.N2')).H(xor(N2', H(R'.xor(H(IDu.PWu), R1').Kser)).
              xor(H(xor(H(IDu.PWu), R').N1'.N2'), xor(H(IDu.PWu), R1')).
              xor(H(R1'.xor(H(IDu.PWu), R1').Kser), H(xor(H(IDu.PWu), R').
              N1'.N2')).N2'.IDsn.Pks.T2').T2') =|>
% USN's acceptance of the values t2 and n2 for USN by SSA
State' := 3  ∧ request(SSA, USN, ssa_usn_t2, T2')
        ∧ request(SSA, USN, ssa_usn_n2, N2')
end role
```

FIGURE 6. HLPSL specification for the role of a user/sensor node.

implemented. The registration takes place via secure channel be means of encrypting the registration messages using a pre-defined secret key, *SKus* between the user and the server. The authentication phase is implemented via public channel. In this phase, the user sends the message $\{PID, A_{SN}, S1, S2, t1\}$ to the *IAP* which is forwarded to the server by the *IAP*. Later, the user receives the message $\{S3, S4, S5, S6, t2\}$ from the *IAP* which was forwarded by the server to the *IAP*.

By the declaration: secret({IDu, PWu, K'}, sp1, {USN}), it is meant that the credentails (user identity ID_U, password PW_U and random secret k) are only known to the user. The declarations: witness(USN, SSA, usn_ssa_t1, T1') and witness(USN, SSA, usn_ssa_n1, N1') state that the user

```
role server(USN, SSA, IAP: agent, H: hash_func, Snd, Rcv: channel(dy))
% Player: Server (SSA)
played_by SSA
def=
local State: nat,
    SKus, SKsap: symmetric_key,
    IDu, PWu, R, K, Kser, IDiap, S, Asn, Bsn, Pks, N1, IDsn: text,
 N2, T1, T2, S1, R1, S3, S4, S5, S6 : text
const sp1, sp2, sp3, usn_ssa_t1, usn_ssa_n1,
    ssa_usn_t2, ssa_usn_n2: protocol_id
init  State := 0

transition
%%% User registration phase
%%% Receive registration information from USN via secure channel
1. State = 0 /\ Rcv({xor(H(IDu.PWu), R').R'}_SKus) =|>
State' := 3  /\ secret({IDu,PWu,K'}, sp1, {USN})
         /\ secret({R'}, sp2, {USN,SSA})
         /\ S' := H(xor(H(IDu.PWu), R').Kser)
         /\ Asn' := xor(R', S')
         /\ Bsn' := H(R'.xor(H(IDu.PWu), R').Kser)
%%% Send registration response to USN via secure channel
         /\ Snd({S'.Asn'.Bsn'}_SKus)
%%% IAP registration phase
%%% Send registration information to IAP via secure channel
         /\ Snd({IDiap}_SKsap)
         /\ secret({IDu,PWu,K'}, sp1, {USN})
         /\ secret({R',Pks}, sp2, {USN,SSA})
         /\ secret({Kser}, sp3, {SSA})
%%% Authentication phase
%%% Receive message from USN/IAP via public channel
2. State = 3 /\ Rcv(xor(H(IDu.PWu), R').xor(R', H(xor(H(IDu.PWu), R').Kser)).
              xor(H(R'.xor(H(IDu.PWu), R').Kser), N1').
              H(IDsn.xor(R', H(xor(H(IDu.PWu), R').Kser)).
              xor(H(R'.xor(H(IDu.PWu), R').Kser), N1').T1'.N1').T1'.IDiap) =|>
State' := 5  /\ N2' := new() /\ T2' := new()
         /\ R1' := H(N1'.N2')
         /\ S3' := xor(N2', H(R'.xor(H(IDu.PWu), R1').Kser))
         /\ S4' := xor(H(xor(H(IDu.PWu), R').N1'.N2'), xor(H(IDu.PWu), R1'))
         /\ S5' := xor(H(R1'.xor(H(IDu.PWu), R1').Kser), H(xor(H(IDu.PWu), R').
                N1'.N2'))
         /\ S6' := H(S3'.S4'.S5'.N2'.IDsn.Pks.T2')
%%% Send message to IAP via public channel
         /\ Snd(S3'.S4'.S5'.S6'.T2'.IDiap)
%% SSA has freshly generated the values t2 and n2 for IAP/USN
        /\ witness(SSA, USN, ssa_usn_t2, T2')
        /\ witness(SSA, USN, ssa_usn_n2, N2')
% SSA's acceptance of the values t1 and n1 for SSA by USN
       /\ request(USN, SSA, usn_ssa_t1, T1')
       /\ request(USN, SSA, usn_ssa_n1, N1')
end role
```

FIGURE 7. HLPSL specification for the role of a server/system administrator.

```
role accesspoint(USN, SSA, IAP: agent, H: hash_func, Snd, Rcv: channel(dy))
% Player: IAP: Intermediate access point
played_by IAP
def=
local State: nat,
    SKsap: symmetric_key,
    IDu, PWu, R, K, Kser, IDiap, Pks, IDsn, N1, T1: text,
 N2, R1, T2: text
const sp1, sp2, sp3 : protocol_id
init  State := 0
transition
%%% IAP registration phase
1. State = 0 /\ Rcv({IDiap}_SKsap)  =|>
State' := 2  /\ secret({IDu,PWu,K'}, sp1, {USN})
         /\ secret({R',Pks}, sp2, {USN,SSA})
         /\ secret({Kser}, sp3, {SSA})
%%% Authentication phase
%%% Receive message from USN via public channel
2. State = 2 /\ Rcv(xor(H(IDu.PWu), R').xor(R', H(xor(H(IDu.PWu), R').Kser)).
              xor(H(R'.xor(H(IDu.PWu), R').Kser), N1').
              H(IDsn.xor(R', H(xor(H(IDu.PWu), R').Kser)).
              xor(H(R'.xor(H(IDu.PWu), R').Kser), N1').T1'.N1').T1') =|>
%%% Forward received message including its identity to SSA via public channel
State' := 4  /\ Snd(xor(H(IDu.PWu), R').xor(R', H(xor(H(IDu.PWu), R').Kser)).
              xor(H(R'.xor(H(IDu.PWu), R').Kser), N1').
              H(IDsn.xor(R', H(xor(H(IDu.PWu), R').Kser)).
              xor(H(R'.xor(H(IDu.PWu), R').Kser), N1').T1'.N1').T1'.IDiap)
%%% Receive message from SSA via public channel
3. State = 4 /\ Rcv(xor(N2', H(R'.xor(H(IDu.PWu), R1').Kser)).
              xor(H(xor(H(IDu.PWu), R').N1'.N2'), xor(H(IDu.PWu), R1')).
              xor(H(R1'.xor(H(IDu.PWu), R1').Kser), H(xor(H(IDu.PWu), R').
              N1'.N2')).H(xor(N2', H(R'.xor(H(IDu.PWu), R1').Kser)).
              xor(H(xor(H(IDu.PWu), R').N1'.N2'), xor(H(IDu.PWu), R1')).
              xor(H(R1'.xor(H(IDu.PWu), R1').Kser), H(xor(H(IDu.PWu), R').
              N1'.N2')).N2'.IDsn.Pks.T2').T2'.IDiap)  =|>
%%% Forward received message excluding its identity to USN via public channel
State' := 6  /\ Snd(xor(N2', H(R'.xor(H(IDu.PWu), R1').Kser)).
              xor(H(xor(H(IDu.PWu), R').N1'.N2'), xor(H(IDu.PWu), R1')).
              xor(H(R1'.xor(H(IDu.PWu), R1').Kser), H(xor(H(IDu.PWu), R').
              N1'.N2')).H(xor(N2', H(R'.xor(H(IDu.PWu), R1').Kser)).
              xor(H(xor(H(IDu.PWu), R').N1'.N2'), xor(H(IDu.PWu), R1')).
              xor(H(R1'.xor(H(IDu.PWu), R1').Kser), H(xor(H(IDu.PWu), R').
              N1'.N2')).N2'.IDsn.Pks.T2').T2')
end role
```

FIGURE 8. HLPSL specification for the role of the intermediate access point.

has freshly generated the values $t1$ and $n1$ for IAP/SSA which are included in the message $\{PID, A_{SN}, S1, S2, t1\}$. By the declarations: request(SSA, USN, ssa_usn_t2, T2') and request(SSA, USN, ssa_usn_n2, N2'), we mean the user's acceptance of the values $t2$ and $n2$ for teh user by the server that were included in the message $\{S3, S4, S5, S6, t2\}$. In the role for "goal and environment", two security goals are achieved: a) "privacy (confidentiality)" and b) "authentication".

C. ANALYSIS OF RESULTS

Under the HLPSL implementation of LAKS-NVT described in Section VII-B, we have simulated LAKS-NVT under both the OFMC and CL-AtSe backends using the widely-used "SPAN, the Security Protocol ANimator for AVISPA" tool [49]. Since AVISPA uses the "Dolev-Yao (DY) threat model" [31], two attacks, namely "replay" and "man-in-the-middle" attacks are detected.

The intruder simulation under the SPAN has been demonstrated in Figure 10. From this figure, it is seen that there are no attacks on the proposed scheme (LAKS-NVT). Finally, the simulation results under the OFMC and CL-AtSe backends are shown in Figure 11. It is also clear that LAKS-NVT is resilient against both "replay" and "man-in-the-middle" attacks.

VIII. PRACTICAL PERSPECTIVE: NS2 SIMULATION STUDY

The practical demonstration of LAKS-NVT using the well-known NS2 simulator [50] is executed in this section. In recent years, NS2 simulator becomes also a popular simulation tool for measuring the network performance parameters in many other networks apart from "simulation of

```
%%% Role for the session
role session (USN, SSA, IAP: agent, H: hash_func)
def=
  local  Snd1, Snd2, Snd3, Rcv1, Rcv2, Rcv3: channel (dy)
  composition
     user(USN, SSA, IAP, H, Snd1, Rcv1)
   /\ server(USN, SSA, IAP, H, Snd2, Rcv2)
   /\ accesspoint(USN, SSA, IAP, H, Snd3, Rcv3)
end role


%%% Role for the goal and environment
role environment()
def=
  const usn, ssa, iap: agent,
        h: hash_func, t1, t2, idiap: text,
        sp1, sp2, sp3, usn_ssa_t1, usn_ssa_n1,
        ssa_usn_t2, ssa_usn_n2: protocol_id
  intruder_knowledge = {usn, ssa, iap, h, t1, t2, idiap}
  composition
          session(usn, ssa, iap, h)
       /\ session(i, ssa, iap, h)
       /\ session(usn, i, iap, h)
/\ session(usn, ssa, i, h)
end role
goal
%%% Confidentiality (privacy)
  secrecy_of sp1, sp2, sp3
%%% Authentication
  authentication_on usn_ssa_t1, usn_ssa_n1
  authentication_on ssa_usn_t2, ssa_usn_n2
end goal
environment()
```

FIGURE 9. HLPSL specification for the roles of the session, goal and environment.

Transmission Control Protocol (TCP), routing, and multicast protocols over wired and wireless networks'' [13], [51], [52].

IX. SIMULATION PARAMETERS

Table 4 consists of the details of various parameters used during the simulation. Ubuntu 18.04 LTS platform was utilized for conducting the simulation with the help of NS2 2.35 simulation tool. The wireless protocol IEEE 802.11 was used. Two different cases were considered in the simulation. We have taken one intermediate access point (IAP) and one server/system administrator for both cases. The number of sensor nodes were taken as 25 (in Case 1) and 30 (in Case 2). The simulation was conducted for a duration of 1800 seconds. The communication ranges of sensor nodes and intermediate access point are considered as 200 and 1000 meters, respectively. The Ad-hoc on-demand distance vector routing (AODV) [55] designed by Perkins and Royer was considered

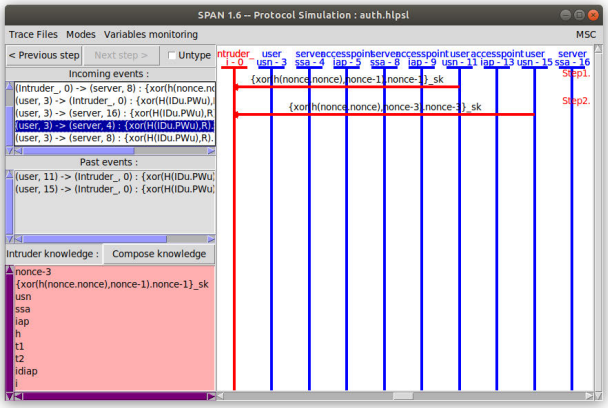

FIGURE 10. Intruder simulation under SPAN.

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS

PROTOCOL
  /home/akdas/Desktop/span/
  testsuite/results/auth.if

GOAL  as specified

BACKEND  OFMC

STATISTICS
  TIME 437 ms
  parseTime 0 ms
  visitedNodes: 320 nodes
  depth: 7 plies
```

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/akdas/Desktop/span/
  testsuite/results/auth.if
GOAL
  As specified

BACKEND
  CL-AtSe

STATISTICS
  Analysed   : 63 states
  Reachable  : 15 states
  Translation: 0.10 seconds
  Computation: 0.01 seconds
```

FIGURE 11. Simulation results under OFMC and CL-AtSe backends.

TABLE 4. Various parameters used during simulation.

Parameter	Description
Platform	Ubuntu 18.04 LTS
Tool used	NS2 2.35
Wireless protocol	802.11
Number of Intermediate access point (for both cases)	1
Number of Server/system administrator (for both cases)	1
Number of sensor nodes	25 (case-1), 30 (case-2)
Simulation time	1800 seconds
Communication range of sensor nodes	200 meters
Communication range of intermediate access point	1000 meters
Routing protocol	AODV [55]

as the routing protocol. The remaining parameters associated with the NS2 simulation were taken with the standard values.

The communication costs between various entities are computed as follows. During the MAKA phase among sensor node (*SN*), intermediate access point (*IAP*) and server/system administrator (S/SA), we have the following messages exchanged among the entities:

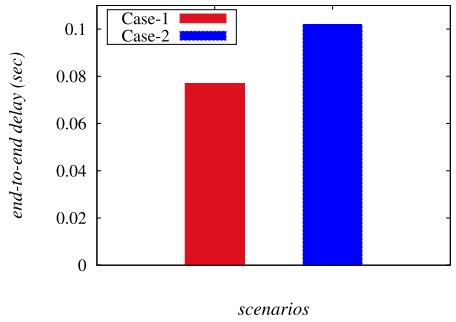

FIGURE 12. End-to-end delay (in seconds) in various scenarios.

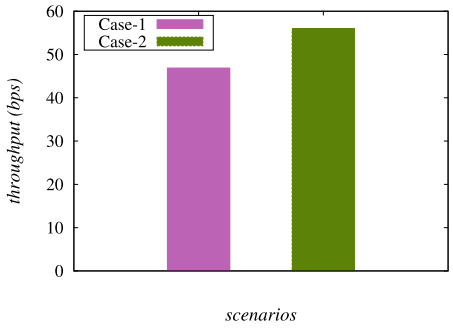

FIGURE 13. Throughput (in bps) in various scenarios.

- The message $Msg_1 = \{PID, A_{SN}, S1, S2, t1\}$ from SN to IAP needs (160+ 160+ 160+ 160+ 32) = 672 bits.
- The message $Msg_2 = \{PID, A_{SN}, S1, S2, t1, ID_{IAP}\}$ from IAP to S/SA requires (160+ 160+ 160+ 160+ 32+ 160) = 832 bits.
- The message $Msg_3 = \{S3, S4, S5, S6, t2, ID_{IAP}\}$ from S/SA to IAP needs (160+ 160+ 160+ 160+ 32+ 160) = 832 bits.
- The message $Msg_4 = \{S3, S4, S5, S6, t2\}$ from IAP to SN demands (160+ 160+ 160+ 160+ 32) = 672 bits.

A. DISCUSSION ON SIMULATION RESULTS

During the experimentation, we have calculated ''the network performance parameters, such as end-to-end delay (in seconds), throughput (in bits per second) and packet loss rate''.

1) IMPACT ON END-TO-END DELAY

The end-to-End Delay (*EED*) is defined as ''the average time required by the messages that reached the destination station from the source station''. It can be mathematically calculated as ''$\sum_{i=1}^{v_p}(T_{R_i} - T_{S_i})/v_p$, where T_{S_i} and T_{R_i} are sending and receiving packet time of i respectively, v_p is the total number of exchanged messages''. In an authentication and key agreement procedure, it is important to calculate the value of *EED*, because it is needed for ''the establishment of session key among the communicating parties with the help of certain exchange of messages''. It is expected that the *EED* value should be less for an efficient authentication and key agreement mechanism. The *EED* values for LAKS-NVT for both considered cases (for instance, Case 1 and Case 2) are depicted in Figure 12. The *EED* values are 0.07697 and 0.10196 seconds for Case 1 and Case 2, respectively. Furthermore, it is worth noticing that the value of *EED* increases with the increasing number of sensor nodes because it causes the increment in the number of exchanged messages. Hence, there is a slight increment in *EED* from Case 1 to Case 2.

2) IMPACT ON THROUGHPUT

Throughput is another essential network performance parameter that can be computed as ''the number of bits transmitted per unit of time''. The throughput (in bps) values of LAKS-NVT for various considered cases are depicted in Figure 13. It can be mathematically formulated as ''$\frac{N_r \times |PKS|}{T_\tau}$, where T_τ is the total time (in seconds), $|PKS|$ packet's size and N_r is total number of received packets''. Furthermore, the considered simulation time was 1800 seconds, which was the total time. The throughput values of LAKS-NVT are 46.88 and 55.91 *bps* for Case 1 and Case 2, receptively. The value of the throughput increases with the increasing number of sensor nodes (for instance, Case 1 to Case 2). This is because there was an increment in the number of exchanged messages from Case 1 to Case 2, which further increases the network throughput.

3) IMPACT ON PACKET LOSS RATE

Packet loss rate is also another crucial network performance parameter which is formulated as the ''number of packets loss per unit time'' and defined by ''$\frac{N_{lp}}{T_d}$, where T_d is the total time (in seconds) and N_{lp} is the total number of lost packets in a given duration of time''. An authentication and key agreement scheme is considered to be reliable if it produces less ''packet loss rate''. The packet loss rates of LAKS-NVT under various cases are depicted in Figure 14. The considered simulation time (total time) is 1800 seconds. The values of ''packet loss rate'' of LAKS-NVT are 0.01556 and 0.01667 for Case 1 and Case 2, respectively. Moreover, the ''packet loss rate'' increases with the increasing number of sensor nodes, because with the increasing number of sensor nodes more number of messages are required to be exchanged. It further causes traffic congestion, and therefore, ''packet loss rate'' also increases from Case 1 to Case 2. However, the increased value of ''packet loss rate'' is marginal as LAKS-NVT utilizes the ''lightweight cryptographic methods''.

4) IMPACT ON PACKET DELIVERY RATIO

It is ''the ratio of number of received packets to number of sent packets''. For an efficient and reliable communication system its value should be closer to 1. The packet delivery ratio of LAKS-NVT under various cases are depicted in Fig. 15. The values of ''packet delivery ratio'' of LAKS-NVT are 0.97 and 0.96 for Case 1 and Case 2, respectively. Moreover, the ''packet delivery ratio decreases slightly with the increasing number of sensor nodes, because

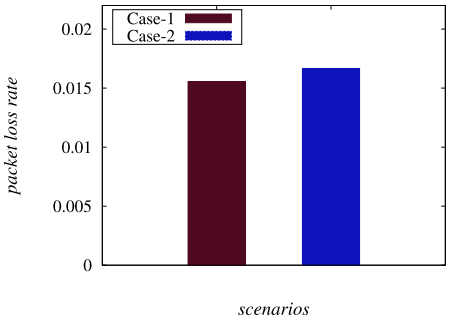

FIGURE 14. Packet loss rate.

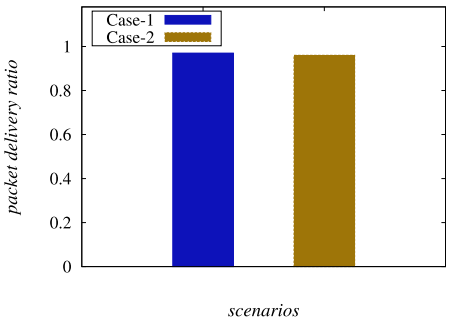

FIGURE 15. Packet delivery ratio in various scenarios.

with the increasing number of sensor nodes more number of messages are required to be exchanged''. It further causes traffic congestion, and therefore, the ''packet delivery ratio also decreases from Case 1 to Case 2''. However, the decreased value of ''packet delivery ratio'' is marginal as LAKS-NVT utilizes the ''lightweight cryptographic methods''.

X. PERFORMANCE ANALYSIS

We compared the propose scheme performance with contemporary authentication schemes for medical IoT [26], [28], [30], and also compared security features to verify LAKS-NVT offers enhanced secure.

A. SECURITY FEATURES

Table 5 compares security features between LAKS-NVT and several contemporary schemes. LAKS-NVT can withstand more potential attacks than any other scheme, and is secure against leaking verification table attack because user sensitive information is not stored in the server database. Therefore, LAKS-NVT is significantly more secure, achieving essential security requirements for medical IoT environments.

B. COMPUTATION AND COMMUNICATION OVERHEADS

For comparative analysis on communication and computational costs, we consider the authentication and key agreement phase for LAKS-NVT and other compared schemes.

Tables 6 and 7 compare computational and communication costs, respectively, between LAKS-NVT and contemporary lightweight authentication schemes. All considered

TABLE 5. Security features for the proposed scheme and existing contemporary schemes.

Feature	Li *et al.* [28]	Xu *et al.* [26]	Xu *et al.* [30]	Proposed
SF_1	×	×	×	○
SF_2	×	×	×	○
SF_3	○	○	○	○
SF_4	×	×	×	○
SF_5	○	○	○	×
SF_6	×	×	×	○
SF_7	○	○	×	○
SF_8	○	×	×	○

○: ''security feature is satisfied''; ×: ''security feature is not satisfied''; SF_1: impersonation attack; SF_2: stolen device attack; SF_3: replay attack; SF_4: formal (mathematical) analysis using ROR model; SF_5: untraceability; SF_6: MA; SF_7: anonymity; SF_8: leaking verification table attack

TABLE 6. Computation overheads.

Scheme	Total computation cost
Li *et al.* [28]	$8T_h \approx 0.004$ s
Xu *et al.* [26]	$10T_h \approx 0.005$ s
Xu *et al.* [30]	$11T_h \approx 0.0055$ s
Proposed	$20T_h \approx 0.01$ s

TABLE 7. Communication overheads.

Scheme	Total communication cost (in bits)
Li *et al.* [28]	2944
Xu *et al.* [26]	2624
Xu *et al.* [30]	2688
Proposed	3008

schemes [26], [28], [30] have high efficiency because they require only hash and XOR operations. Table 6 only considers the hash operation, since XOR computational costs are negligible.

We use exchanged message sizes to analyze communication overheads. Hash function (if SHA-1 hash function is applied), random number, and identity are all 160 bits [53], and timestamp is taken as 32 bits. In LAKS-NVT, the messages $\{PID, A_{SN}, S1, S2, t1\}$, $\{PID, A_{SN}, S1, S2, t1, ID_{IAP}\}$, $\{S3, S4, S5, S6, t2, ID_{IAP}\}$ and $\{S3, S4, S5, S6, t2\}$ require $(160 + 160 + 160 + 160 + 32) = 672$ bits, $(160 + 160 + 160 + 160 + 32 + 160) = 832$ bits, $(160 + 160 + 160 + 160 + 32 + 160) = 832$ bits, and $(160 + 160 + 160 + 160 + 32) = 672$ bits, respectively. Thus, the total communication cost requires for LAKS-NVT becomes $(672 + 832 + 832 + 672) = 3008$ bits. As a result, LAKS-NVT incurs the communication cost of 3008 bits, whereas the schemes of Li *et al.* [28], Xu *et al.* [26], and Xu *et al.* [30] incur the communication costs of 2944 bits, 2624 bits and 2688 bits, respectively.

For computational costs comparison, T_h denotes the time needed for a ''cryptographic one-way hash function''. Based on the experimental results reported in [53], [54], we consider $T_h \approx 0.5$ ms. LAKS-NVT needs the computation cost 20 $T_h \approx 0.01$ seconds, whereas the computation cost for other schemes, such as the schemes of Li *et al.* [28], Xu *et al.* [26] and Xu *et al.* [30] require 8 $T_h \approx 0.004$ seconds, 10 $T_h \approx 0.005$ seconds and 11 $T_h \approx 0.0055$ seconds, respectively.

Section X-A showed that the schemes [26], [28], [30] are unsuitable for practical environments because they are vulnerable to various attacks, including impersonation, stolen device and leaking verification attacks. Thus, although LAKS-NVT has slightly higher computational cost than the considered contemporary schemes, it is significantly more secure and also provides session key security. Therefore, LAKS-NVT can successfully protect user privacy in the practical medical IoT environments.

XI. CONCLUSION

This paper proved that the previous Xu *et al.*'s scheme does not prevent various attacks, including impersonation, stolen SN, and leaking verification table attacks; and does not achieve anonymity, secure MA, and untraceability. To overcome these security flaws, we designed a provably secure and lightweight MAKA scheme for medical IoT without requiring a server verification table.

We showed LAKS-NVT was secure against impersonation, stolen SN, replay, and leaking verification table attacks since it does not store user sensitive data in a server database. LAKS-NVT also achieves anonymity, secure MA and untraceability. Formal (mathematical) security analysis confirmed that LAKS-NVT guaranteed secure MA between SN and S using the ''BAN logic, and session key security using the ROR model''. In addition, the formal security verification using the AVISPA tool proves that LAKS-NVT is also secure.

Performance comparison with contemporary lightweight authentication schemes confirmed that computational and communication cost performances were comparable with the contemporary schemes. Furthermore, LAKS-NVT exhibited significantly enhanced security and functionality. In addition, through the NS2 simulation study we evaluated the network performance of LAKS-NVT. Therefore, LAKS-NVT is suitable for practical medical IoT environments.

ACKNOWLEDGEMENT

The authors thank the anonymous reviewers and associate editor for their valuable feedback on the paper, which helped them to improve its quality and presentation.

REFERENCES

[1] R. van der Meulen and V. Woods, *More Than Half of Major New Business Processes and Systems Will Incorporate Some Element of the Internet of Things*, Gartner, Stamford, CT, USA, 2016. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2016-01-14-gartner-says-by-2020-more-than-half-of-major-new-business-processes-and-systems-will-incorporate-some-element-of-the-internet-of-things

[2] T. G. Zimmerman, ''Personal area networks: Near-field intrabody communication,'' *IBM Syst. J.*, vol. 35, no. 3.4, pp. 609–617, 1996.

[3] L. Lamport, ''Password authentication with insecure communication,'' *Commun. ACM*, vol. 24, no. 11, pp. 770–772, Nov. 1981.

[4] M. L. Das, A. Saxena, and V. P. Gulati, ''A dynamic ID-based remote user authentication scheme,'' *IEEE Trans. Consum. Electron.*, vol. 50, no. 2, pp. 629–631, May 2004.

[5] K. Park, Y. Park, Y. Park, A. Goutham Reddy, and A. K. Das, ''Provably secure and efficient authentication protocol for roaming service in global mobility networks,'' *IEEE Access*, vol. 5, pp. 25110–25125, 2017.

[6] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, and N. Kumar, ''A robust and anonymous patient monitoring system using wireless medical sensor networks,'' *Future Gener. Comput. Syst.*, vol. 80, pp. 483–495, Mar. 2018.

[7] S. Kumari, M. K. Khan, and X. Li, ''A more secure digital rights management authentication scheme based on smart card,'' *Multimedia Tools Appl.*, vol. 75, no. 2, pp. 1135–1158, Jan. 2016.

[8] C. Wang, D. Wang, G. Xu, and Y. Guo, ''A lightweight password-based authentication protocol using smart card,'' *Int. J. Commun. Syst.*, vol. 30, no. 16, p. e3336, Nov. 2017.

[9] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, ''A secure authentication protocol for Internet of vehicles,'' *IEEE Access*, vol. 7, pp. 12047–12057, 2019.

[10] A. K. Das, ''A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor,'' *Int. J. Commun. Syst.*, vol. 30, no. 1, p. e2933, Jan. 2017.

[11] Y. Chen, Y. Ge, Y. Wang, and Z. Zeng, ''An improved three-factor user authentication and key agreement scheme for wireless medical sensor networks,'' *IEEE Access*, vol. 7, pp. 85440–85451, 2019.

[12] J. Lee, S. Yu, K. Park, Y. Park, and Y. Park, ''Secure three-factor authentication protocol for multi-gateway IoT environments,'' *Sensors*, vol. 19, no. 13, pp. 2358–2382, May 2019.

[13] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. Goutham Reddy, E.-J. Yoon, and K.-Y. Yoo, ''Secure signature-based authenticated key establishment scheme for future IoT applications,'' *IEEE Access*, vol. 5, pp. 3028–3043, 2017.

[14] M. Wazid, A. K. Das, R. Hussain, G. Succi, and J. J. P. C. Rodrigues, ''Authentication in cloud-driven IoT-based big data environment: Survey and outlook,'' *J. Syst. Archit.*, vol. 97, pp. 185–196, Aug. 2019.

[15] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, ''Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment,'' *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4900–4913, Dec. 2018.

[16] V. Odelu, A. K. Das, M. Khurram Khan, K.-K.-R. Choo, and M. Jo, ''Expressive CP-ABE scheme for mobile devices in IoT satisfying constant-size keys and ciphertexts,'' *IEEE Access*, vol. 5, pp. 3273–3283, 2017.

[17] A. K. Das, M. Wazid, A. R. Yannam, J. J. P. C. Rodrigues, and Y. Park, ''Provably secure ECC-based device access control and key agreement protocol for IoT environment,'' *IEEE Access*, vol. 7, pp. 55382–55397, 2019.

[18] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, ''Secure remote user authenticated key establishment protocol for smart home environment,'' *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 2, pp. 391–406, Mar. 2020.

[19] M. Wazid, A. K. Das, N. Kumar, V. Odelu, A. Goutham Reddy, K. Park, and Y. Park, ''Design of lightweight authentication and key agreement protocol for vehicular ad hoc networks,'' *IEEE Access*, vol. 5, pp. 14966–14980, 2017.

[20] S. Chatterjee, A. K. Das, and J. K. Sing, ''An enhanced access control scheme in wireless sensor networks,'' *Ad Hoc Sensor Wireless Netw.*, vol. 21, nos. 1–2, pp. 121–149, Jan. 2014.

[21] M. Wazid, A. K. Das, S. Kumari, X. Li, and F. Wu, ''Design of an efficient and provably secure anonymity preserving three-factor user authentication and key agreement scheme for TMIS,'' *Secur. Commun. Netw.*, vol. 9, no. 13, pp. 1983–2001, Sep. 2016.

[22] J. Srinivas, A. K. Das, N. Kumar, and J. Rodrigues, ''Cloud centric authentication for wearable healthcare monitoring system,'' *IEEE Trans. Dependable Secure Comput.*, early access, Apr. 19, 2018, doi: 10.1109/TDSC.2018.2828306.

[23] V. Odelu, A. K. Das, and A. Goswami, ''SEAP: Secure and efficient authentication protocol for NFC applications using pseudonyms,'' *IEEE Trans. Consum. Electron.*, vol. 62, no. 1, pp. 30–38, Feb. 2016.

[24] J. Liu, Z. Zhang, X. Chen, and K. S. Kwak, ''Certificateless remote anonymous authentication schemes for WirelessBody area networks,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 332–342, Feb. 2014.

[25] Z. Zhao, ''An efficient anonymous authentication scheme for wireless body area networks using elliptic curve cryptosystem,'' *J. Med. Syst.*, vol. 38, no. 2, pp. 1–7, Jan. 2014.

[26] Z. Xu, C. Xu, H. Chen, and F. Yang, ''A lightweight anonymous mutual authentication and key agreement scheme for WBAN,'' *Concurrency Comput., Pract. Exper.*, vol. 31, no. 14, Jul. 2019.

[27] M. H. Ibrahim, S. Kumari, A. K. Das, M. Wazid, and V. Odelu, "Secure anonymous mutual authentication for star two-tier wireless body area networks," *Comput. Methods Programs Biomed.*, vol. 135, pp. 37–50, Oct. 2016.

[28] X. Li, M. H. Ibrahim, S. Kumari, A. K. Sangaiah, V. Gupta, and K.-K.-R. Choo, "Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks," *Comput. Netw.*, vol. 129, pp. 429–443, Dec. 2017.

[29] X. Li, J. Niu, S. Kumari, F. Wu, and K.-K.-R. Choo, "A robust biometrics based three-factor authentication scheme for global mobility networks in smart city," *Future Gener. Comput. Syst.*, vol. 83, pp. 607–618, Jun. 2018.

[30] Z. Xu, C. Xu, W. Liang, J. Xu, and H. Chen, "A lightweight mutual authentication and key agreement scheme for medical Internet of Things," *IEEE Access*, vol. 7, pp. 53922–53931, 2019.

[31] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[32] M. Abdalla, P. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. 8th Int. Workshop Theory Pract. Public Key Cryptogr. (PKC)*, Les Diablerets, Switzerland, 2005, pp. 65–84.

[33] S. Barman, A. K. Das, D. Samanta, S. Chattopadhyay, J. J. P. C. Rodrigues, and Y. Park, "Provably secure multi-server authentication protocol using fuzzy commitment," *IEEE Access*, vol. 6, pp. 38578–38594, 2018.

[34] K. Park, Y. Park, A. K. Das, S. Yu, J. Lee, and Y. Park, "A dynamic privacy-preserving key management protocol for V2G in social Internet of Things," *IEEE Access*, vol. 7, pp. 76812–76832, 2019.

[35] A. K. Das, M. Wazid, N. Kumar, M. K. Khan, K.-K.-R. Choo, and Y. Park, "Design of secure and lightweight authentication protocol for wearable devices environment," *IEEE J. Biomed. Health Informat.*, vol. 22, no. 4, pp. 1310–1322, Jul. 2018.

[36] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.

[37] AVISPA. (2019). *Automated Validation of Internet Security Protocols and Applications*. Accessed: Oct. 2019. [Online]. Available: http://www.avispa-project.org/

[38] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Netw.*, vol. 20, pp. 96–112, Sep. 2014.

[39] C.-C. Chang and H.-D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 357–366, Jan. 2016.

[40] X. Li, J. Peng, J. Niu, F. Wu, J. Liao, and K.-K.-R. Choo, "A robust and energy efficient authentication protocol for industrial Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1606–1615, Jun. 2018.

[41] P. Gope and T. Hwang, "An efficient mutual authentication and key agreement scheme preserving strong anonymity of the mobile user in global mobility networks," *J. Netw. Comput. Appl.*, vol. 62, pp. 1–8, Feb. 2016.

[42] S. Janbabaei, H. Gharaee, and N. Mohammadzadeh, "Lightweight, anonymous and mutual authentication in IoT infrastructure," in *Proc. 8th Int. Symp. Telecommun. (IST)*, Sep. 2016, pp. 162–166.

[43] D.-H. Hwang, J.-M. Shin, and Y.-H. Choi, "Authentication protocol for wearable devices using mobile authentication proxy," in *Proc. 10th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2018, pp. 700–702.

[44] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, May 2002.

[45] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA, 1999, pp. 388–397.

[46] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2776–2791, Nov. 2017, doi: 10.1109/TIFS.2017.2721359.

[47] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 4, pp. 428–442, Jul. 2015.

[48] (2006). *Automated Validation of Internet Security Protocols and Applications*. Accessed: Mar. 2020. [Online]. Available: http://www.avispa-project.org/

[49] (2020). *SPAN, the Security Protocol ANimator for AVISPA*. Accessed: Mar. 2020. [Online]. Available: http://www.avispa-project.org/

[50] *The Network Simulator-ns-2*. Accessed: Apr. 2020. [Online]. Available: https://www.isi.edu/nsnam/ns/

[51] M. Wazid, A. K. Das, N. Kumar, and A. V. Vasilakos, "Design of secure key management and user authentication scheme for fog computing services," *Future Gener. Comput. Syst.*, vol. 91, pp. 475–492, Feb. 2019.

[52] M. Wazid, P. Bagga, A. K. Das, S. Shetty, J. J. P. C. Rodrigues, and Y. Park, "AKM-IoV: Authenticated key management protocol in fog computing-based Internet of vehicles deployment," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8804–8817, Oct. 2019.

[53] P. Chandrakar and H. Om, "A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ECC," *Comput. Commun.*, vol. 110, pp. 26–34, Sep. 2017.

[54] C.-T. Li, M.-S. Hwang, and Y.-P. Chu, "A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks," *Comput. Commun.*, vol. 31, no. 12, pp. 2803–2814, Jul. 2008.

[55] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. 2nd IEEE Workshop Mobile Comput. Syst. Appl. (WMCSA)*, New Orleans, LA, USA, Feb. 1999, pp. 90–100.

[56] C.-M. Chen, K.-H. Wang, K.-H. Yeh, B. Xiang, and T.-Y. Wu, "Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 8, pp. 3133–3142, Aug. 2019.

KISUNG PARK received the B.S. and M.S. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the School of Electronics Engineering. He is currently a Researcher with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. His research interests include authentication, blockchain, anonymous credentials, decentralized identifier, the Internet of Things, post-quantum cryptography, VANET, and information security.

SUNGKEE NOH received the M.S. degree from Postech, South Korea, in 1992, and the Ph.D. degree from Chungnam National University, South Korea, in 2004. He is currently the Principal Researcher of the Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. His current research interests include the Internet of Things, decentralized identifier, and blockchain.

HYUNJIN LEE received the B.S. and M.S. degrees in electronic engineering from Kyungpook National University, Daegu, South Korea, in 1997 and 1999, respectively. He is currently the Principal Researcher of the Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. His research interests include blockchain, self-sovereign identity, attribute based credentials, and convergence service control.

ASHOK KUMAR DAS (Senior Member, IEEE) received the M.Sc. degree in mathematics, the M.Tech. degree in computer science and data processing, and the Ph.D. degree in computer science and engineering from IIT Kharagpur, India. He is currently an Associate Professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. His current research interests include cryptography, wireless sensor network security, hierarchical access control, security in vehicular ad hoc networks, smart grid, the Internet of Things (IoT), cyber-physical systems (CPS) and cloud computing, and remote user authentication. He has authored over 225 papers in international journals and conferences in the above areas, including over 190 reputed journal articles. Some of his research findings are published in top cited journals, such as the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON SMART GRID, the IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, the IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS (formerly the IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE), the *IEEE Consumer Electronics Magazine*, IEEE ACCESS, the *IEEE Communications Magazine*, *Future Generation Computer Systems*, *Computers and Electrical Engineering*, *Computer Methods and Programs in Biomedicine*, *Computer Standards and Interfaces*, *Computer Networks*, *Expert Systems with Applications*, and the *Journal of Network and Computer Applications*. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He also served as one of the Technical Program Committee Chairs of the International Congress on Blockchain and Applications (BLOCKCHAIN'19), Avila, Spain, June 2019. He is on the editorial board of the *KSII Transactions on Internet and Information Systems*, the *International Journal of Internet Technology and Secured Transactions* (Inderscience), and *IET Communications*. He is a Guest Editor of *Computers and Electrical Engineering* (Elsevier) for the Special Issue on Big Data and IoT in E-Healthcare and for ICT Express (Elsevier) for the Special Issue on Blockchain Technologies and Applications for 5G Enabled IoT, and has served as a Program Committee Member in many international conferences.

MYEONGHYUN KIM received the B.S. degree in electronics engineering from Kyungpook National University, Daegu, South Korea, in 2018, where he is currently pursuing the M.S. degree with the School of Electronics Engineering. His research interests include authentication, blockchain, the Internet of Things, and information security.

YOUNGHO PARK (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic engineering from Kyungpook National University, Daegu, South Korea, in 1989, 1991, and 1995, respectively. From 1996 to 2008, he was a Professor with the School of Electronics and Electrical Engineering, Sangju National University, South Korea. From 2003 to 2004, he was a Visiting Scholar with the School of Electrical Engineering and Computer Science, Oregon State University, USA. He is currently a Professor with the School of Electronics Engineering, Kyungpook National University. His research interests include computer networks, multimedia, and information security.

MOHAMMAD WAZID (Senior Member, IEEE) received the M.Tech. degree in computer network engineering from Graphic Era University, Dehradun, India, and the Ph.D. degree in computer science and engineering from the International Institute of Information Technology, Hyderabad, India. He is currently working as an Associate Professor with the Department of Computer Science and Engineering, Graphic Era University. Prior to this, he was working as an Assistant Professor with the Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India. He was also a Postdoctoral Researcher with the Cyber Security and Networks Laboratory, Innopolis University, Innopolis, Russia. His current research interests include security, remote user authentication, the Internet of Things (IoT), and cloud computing. He has published more than 60 papers in international journals and conferences in the above areas. He was a recipient of the University Gold Medal and the Young Scientist Award by UCOST, Department of Science and Technology, Government of Uttarakhand, India. He has also received the recognition of the Best Reviewer of 2019 from *ICT Express* (Elsevier) journal.