IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 2, 2017, accepted February 15, 2017, date of publication March 1, 2017, date of current version March 28, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2676119

Secure Signature-Based Authenticated Key Establishment Scheme for Future IoT Applications

SRAVANI CHALLA[1], MOHAMMAD WAZID[1], ASHOK KUMAR DAS[1], NEERAJ KUMAR[2], (Member, IEEE), ALAVALAPATI GOUTHAM REDDY[3], EUN-JUN YOON[4], AND KEE-YOUNG YOO[3], (Member, IEEE)

[1]Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India
[2]Department of Computer Science and Engineering, Thapar University, Patiala 147004, India
[3]School of Computer Science and Engineering, Kyungpook National University, Daegu 702-701, South Korea
[4]Department of Cyber Security, Kyungil University, Gyeongsan 38428, Korea

Corresponding author: E.-J. Yoon (ejyoon@kiu.kr)

This work was supported in part by the BK21 Plus Project entitled SW Human Resource Development Program for Supporting Smart Life within the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, South Korea, under Grant 21A20131600005, in part by the Ministry of Culture, Sports and Tourism and from the Korea Copyright Commission in 2016 under Grant 2016-CCP-9500, and in part by the Basic Science Research Program within the Ministry of Education through the National Research Foundation of Korea under Grant NRF- 2015R1D1A1A01060801 and Grant NRF-2015R1A2A2A01006824.

ABSTRACT Internet of Things (IoT) is a network of all devices that can be accessed through the Internet. These devices can be remotely accessed and controlled using existing network infrastructure, thus allowing a direct integration of computing systems with the physical world. This also reduces human involvement along with improving accuracy and efficiency, resulting in economic benefit. The devices in IoT facilitate the day-to-day life of people. However, the IoT has an enormous threat to security and privacy due to its heterogeneous and dynamic nature. Authentication is one of the most challenging security requirements in the IoT environment, where a user (external party) can directly access information from the devices, provided the mutual authentication between user and devices happens. In this paper, we present a new signature-based authenticated key establishment scheme for the IoT environment. The proposed scheme is tested for security with the help of the widely used Burrows–Abadi–Needham logic, informal security analysis, and also the formal security verification using the broadly accepted automated validation of Internet security protocols and applications tool. The proposed scheme is also implemented using the widely accepted NS2 simulator, and the simulation results demonstrate the practicability of the scheme. Finally, the proposed scheme provides more functionality features, and its computational and communication costs are also comparable with other existing approaches.

INDEX TERMS Internet of things (IoT), authentication, key establishment, Burrows-Abadi-Needham (BAN) logic, AVISPA, NS2 simulation, security.

I. INTRODUCTION

IoT encompasses a system of physical objects that are interconnected to exchange and collect data over the internet. These objects are equipped with the required processing and communication abilities and possess a locatable Internet Protocol address (IP address). The objective here is to integrate computer-based systems and the physical world for economic benefit and to improve accuracy and efficiency while reducing human involvement. Cyber-physical systems such as smart grids and intelligent transportation can be considered as subsets of IoT [1]. The connectivity provided should be beyond machine-to-machine communication covering various protocols and applications interconnecting systems, devices and services. Multiple technologies like wireless communication, embedded systems, machine learning, etc. are the building blocks of this vision. Applications of IoT are diverse including infrastructure management in high-risk conditions, disaster management through environmental monitoring and providing remote health-care services, to list a few. IoT, while broadening access to information, has an enormous threat to security and privacy due to its heterogeneous and dynamic nature. Cyber attacks could change from virtual to physical with the increase in number of wearable devices. An estimated 50 billion objects will be a part of IoT by 2020 [2]. IoT being a relatively new concept, the security challenges involved have not been addressed appropriately at the design level for these objects. Employing effective security practices, especially authentication and key management schemes to protect anonymity and privacy, is required.

A. SYSTEM MODELS

In this paper, we have followed two models which are discussed below.

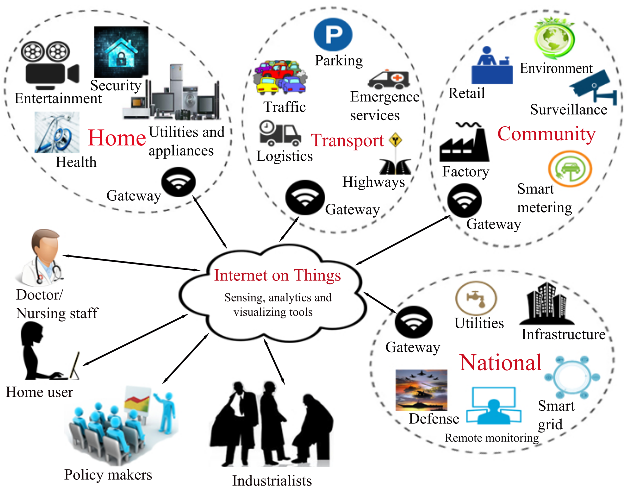

FIGURE 1. Authentication model for IoT applications (Adapted from [2]).

1) IoT AUTHENTICATION MODEL

In the given IoT authentication model shown in Fig. 1, we consider four different scenarios, i.e., Home, Transport, Community and National. All these scenarios have smart devices, such as sensors and actuators. These devices facilitate the day to day life of people. In the given scenarios, all smart devices are connected to the Internet through the gateway nodes (GWNs). Different types of users (for example, smart home user and doctor) can access the data of relevant IoT devices through the GWN. Mutual authentication between a user and a device through the GWN provides access to device data to the user [2].

2) THREAT MODEL

We follow the widely-accepted Dolev-Yao threat (DY) model [3]. Under the DY model, communication between two entities is performed over a public channel. An adversary can then have an opportunity to eavesdrop, modify or delete the content of the messages being transmitted. It is further assumed that the adversary can physically capture one or more sensing devices in IoT, and can extract all the sensitive information stored in the captured devices using the power analysis attacks [4], [5].

B. OUR CONTRIBUTION

The contributions of this paper are:

- An authentication model for IoT is presented and the security challenges involved and its requirements are discussed.
- A secure signature-based authentication and key agreement scheme has been proposed to address these issues.
- A formal security analysis using BAN logic and an informal security analysis have been presented to prove that the scheme is secure.
- Simulation using the AVISPA tool for the formal verification of the scheme's security has also been provided.
- Using NS2 simulator, the scheme's impact on network performance parameters has been measured for practical demonstration of the scheme.
- Finally, it has been shown that the scheme is also efficient in terms of communication and computation costs.

C. ORGANIZATION OF THE PAPER

The paper is organized as follows. In Section II, we discuss the necessary mathematical preliminaries which are needed to describe and analyze the proposed scheme. Section III discusses some security challenges and requirements in IoT. In Section IV, we discuss some existing related work done to address these issues. Sections V and VI present the proposed scheme and its rigorous security analysis, respectively. A comparative analysis of communication and computation costs and functionality features among some related existing schemes for IoT is presented in VII. Section VIII provides an insight into the impact of the scheme on network performance parameters using the NS2 simulator. Finally, some conclusions are drawn in IX.

II. MATHEMATICAL PRELIMINARIES

In this section, we briefly discuss the properties of an elliptic curve over a finite field.

Suppose $a \in Z_p$ and $b \in Z_p$ be two constants, where $Z_p = \{0, 1, \ldots, p-1\}$ and $p > 3$ is a prime. A non-singular elliptic curve $y^2 = x^3 + ax + b$ over the finite field $GF(p)$ is the set $E_p(a, b)$ of the solutions $(x, y) \in Z_p \times Z_p$ to the congruence

$$y^2 \equiv x^3 + ax + b \pmod{p},$$

where $a, b \in Z_p$ such that $4a^3 + 27b^2 \neq 0 \pmod{p}$, with a point at infinity or zero point $\mathcal{O}$.

Let $P = (x_P, y_P) \in E_p(a, b)$ and $Q = (x_Q, y_Q) \in E_p(a, b)$ be two points. Then $x_Q = x_P$ and $y_Q = -y_P$ when $P + Q = \mathcal{O}$. $Q = -P \in E_p(a, b)$ is called the inverse of $P \in E_p(a, b)$. Also, $P + \mathcal{O} = \mathcal{O} + P = P$, for all $P \in E_p(a, b)$. Hasse's theorem states that the number of points on curve $E_p(a, b)$, denoted as $\#E$, satisfies the following inequality [6]:

$$p + 1 - 2\sqrt{p} \leq \#E \leq p + 1 + 2\sqrt{p}.$$

In other words, there are about p points on an elliptic curve $E_p(a, b)$. In addition, $E_p(a, b)$ forms a commutative or an abelian group under addition modulo p operation with $\mathcal{O}$ as the additive identity and $-P \in E_p(a, b)$ as the additive inverse of the point $P \in E_p(a, b)$.

A. ELLIPTIC CURVE POINT ADDITION

Suppose G is the base point on $E_p(a, b)$ with order n, that is, $nG = G+G+\ldots+G(n \text{ times}) = \mathcal{O}$. Let $P, Q \in E_p(a, b)$ be two points on the elliptic curve. Then, $R = (x_R, y_R) = P + Q$ is calculated as follows [6]:

$$x_R = (\lambda^2 - x_P - x_Q) \pmod p,$$
$$y_R = (\lambda(x_P - x_R) - y_P) \pmod p,$$
$$\text{where } \lambda = \begin{cases} \dfrac{y_Q - y_P}{x_Q - x_P} \pmod p, & \text{if } P \neq Q \\ \dfrac{3{x_P}^2 + a}{2y_P} \pmod p, & \text{if } P = Q. \end{cases}$$

B. ELLIPTIC CURVE POINT SCALAR MULTIPLICATION

The elliptic curve multiplication is done as repeated additions. For example, $5P = P + P + P + P + P$ where $P \in E_p(a, b)$.

III. SECURITY CHALLENGES AND REQUIREMENTS IN IoT APPLICATIONS

As accessibility and global connectivity are the key requirements of any IoT application, it increases the available avenues of threats and attacks. The heterogeneous nature of IoT further raises complexity in the deployment of security mechanisms. The wireless nature of most involved entities and their limited capacity are also problematic. Possible transient and random failures are vulnerabilities that attackers could exploit. The various possible attacks on IoT applications are as follows:

- *Denial-of-Service:* Apart from conventional denial-of-service (DoS) attacks like exhausting resources and bandwidth, IoT can be susceptible to attacks on communication infrastructure like channel jamming. Adversaries who are privileged insiders can gain control of the relevant infrastructure to cause more chaos in the network.
- *Controlling:* Active attackers can gain partial or full control of IoT entities and the extent of damage that can be caused is based on the following:
  – Services being provided by the entity.
  – Relevance of the data being managed by that entity.
- *Eavesdropping:* This is a passive attack through which information can be gathered from channel communication. A malicious insider attacker can also gain more advantage by capturing infrastructure or entities.
- *Physical damage:* The easy accessibility of IoT entities and applications can be exploited by attackers to cause physical harm hindering services by attacking an entity or the hardware of the module creating it virtually. Attackers lacking technical knowledge and wanting to cause considerable damage can utilize this.
- *Node capture:* Easy accessibility can also be a vulnerability for information extraction through capturing entities and trying to extract stored data. This is a major threat against data processing and storage entities.

The countermeasures to recover from such attacks once they are detected and diagnosed should be lightweight due to the limited capacity of the involved entities. The solutions must be real-time in nature and if possible, a part of self-healing infrastructure. Any programming information required to deploy the solution should be communicated securely to the entities. The following are some requirements for IoT to counter security breaches:

- *Reliability:* The aim is to guarantee information availability while efficiently managing data storage. Providing redundancy among communication channels through multiple paths is one way to ensure availability.
- *Responsibility:* Otherwise known as access control, this ensures legitimate access to services by defining privacy constraints. The rules for each entity and possible liabilities must be clearly defined to avoid damages.
- *Privacy:* Owing to the ubiquitous nature of IoT, providing privacy is very important. There are the following three areas where privacy has to be ensured:
  – Data sharing and management: This can be achieved by enumerating data aggregated at the sensors. Also, privacy-preservation techniques can be used.
  – Data collection: Some cryptographic approaches mentioned in [7] and [8] can be used.
  – Data security: This can be ensured through password protection.
- *Trust:* IoT being dynamic and distributed, ensuring trust among interacting entities is important. In a heterogeneous network like IoT where devices and not just humans can be involved in trust management, resource constraints should also be considered while developing techniques.
- *Safety:* System components can be prone to sudden failures and safety is required to reduce damage possibilities.
- *Identification and authentication:* Privacy and secure access can be ensured primarily through this. As global access is a necessity in IoT, entities could have one permanent and several temporary identities.

IV. RELATED WORK

Authentication schemes for IoT networks should take into account their dynamic, heterogeneous and distributed nature. These schemes can be broadly classified into categories as follows:

- *Asymmetric key based approach:* Although public key cryptography (for example, RSA algorithm) is suitable for multicast and broadcast, the high communication, computation and storage overheads make it unsuitable for resource constrained applications and networks. Developments in wireless technology have necessitated the implementation of schemes in this category [9]–[12]. Of these, certificate based schemes have unusually heavy overheads. For IoT applications, Datagram

Transport Layer Security (DTLS) based authentication handshake has been proposed in [7]. To counter the high energy consumption due to RSA based encryption and public-key infrastructure certificates in [7], the authors in [13] proposed an elliptic curve cryptography (ECC) based approach. The schemes based on the Merkle hash tree [8] have the advantage of balancing communication and storage overheads, but they are not scalable. Protocols based on user identity [8]–[14] typically use bilinear pairing adding to energy costs. ECC-based RFID authentication schemes are susceptible to tracking attacks on the RFID tag [15]. However, recent research demonstrates that ECC based public key cryptosystem is suitable for resource-constrained devices (for example, sensor nodes in a sensor network) [16]–[19] as only 160-bit ECC offers the same level of security as compared to that of 1024-bit RSA. Thus, ECC is very efficient as compared to RSA due to its smaller key size.

- *Symmetric key based approach:* μTESLA and related schemes [8], [20]–[23] are some of the earliest proposed protocols in this category. Despite reducing energy consumption by using hash functions, they are susceptible to denial-of-service attacks because of delayed authentication, and also do not check for data integrity. Other symmetric key schemes based on key ring [24] and knowledge of deployment [24] are not scalable, and therefore, these are unsuitable for dynamic environments like IoT.
- *Signature based approach:* Schemes similar to [25] and [26] pro vide fast generation and verification of signatures. Also, immediate authentication is guaranteed, and no synchronization is needed. However, long signature and key lengths as used in [27]–[29] make these suitable only for applications that send messages infrequently.

V. PROPOSED SCHEME

In this section, we present a new signature-based authenticated key establishment scheme using the authentication model for IoT applications provided in Fig. 1. As shown in this figure, different users communicate with each other and with various smart devices through gateways to ensure secure communication. The proposed scheme can be applied in all kinds of the IoT applications. For example, a doctor can remotely monitor a patient's vitals through the readings recorded by sensing devices in wireless body area networks. A home user can detect any intrusion by monitoring smart meter readings. In the proposed scheme, a legal user can access the information from a sensing device in the IoT applications provided that both mutually authenticate each other. After their mutual authentication, a secret session key will be established between them for their future secure communications.

The notations used in detailing the proposed scheme have been listed in Table 1. To protect the proposed scheme from strong replay attack, we use both random numbers as well as current timestamps. For this reason, we assume that all the entities involved in IoT environment are synchronized with their clocks. The proposed scheme consists of the following eight phases, namely, 1) system setup, 2) sensing device registration, 3) user registration, 4) login, 5) authentication and key agreement, 6) password & biometric update, 7) smart card revocation and 8) dynamic sensing device addition. The detailed descriptions of these phases are discussed in the following subsections.

TABLE 1. Notations used in this paper.

Symbol	Description
GWN	Gateway node
SD_j	j^{th} sensing device
ID_j	SD_j's identity
U_i	i^{th} user
SC_i	U_i's smart card
ID_i	U_i's identity
PW_i	U_i's password
BIO_i	U_i's personal biometrics template
σ_i	Biometric secret key
τ_i	Biometric public reproduction parameter
t	Error tolerance threshold used by fuzzy extractor
$Gen(\cdot)$	Probabilistic generation procedure used by fuzzy extractor
$Rep(\cdot)$	Deterministic reproduction procedure used by fuzzy extractor
$h(\cdot)$	Collision-resistant one-way cryptographic hash function
p	A large prime number
Z_p	$Z_p = \{0, 1, \cdots, p-1\}$, a prime finite field
E_p	An elliptic curve over prime field Z_p
$P = ((P)_x, (P)_y)$	an elliptic curve point in elliptic curve E_p, $(P)_x$ and $(P)_y$ are x and y coordinates of P, respectively
$k.P$	Elliptic curve point multiplication; $k \in Z_p^*$ being a scalar and $P \in E_p$
d	private key of involved entities
Q	$Q = d.P$, public key of involved entities
T_i, T_s	Current system timestamps
ΔT	Maximum transmission delay
sk_{ij}	Session key between U_i and SD_j
$\oplus$, $\|\|$	Bitwise XOR and concatenation operations, respectively

A. SYSTEM SETUP PHASE

The system setup is done by the gateway node *GWN* as follows.

- **Step S1.** *GWN* chooses a non-singular elliptic curve E_p over a prime finite field Z_p, p being a large prime. *GWN* then selects a base point P of order n over E_p such that $n.P = \mathcal{O}$, where $\mathcal{O}$ is called the point at infinity or zero point. *GWN* also chooses its private key d_{GWN} and computes the corresponding public key $Q_{GWN} = d_{GWN}.P$.
- **Step S2.** *GWN* then chooses a collision-resistant one-way cryptographic hash function $h(\cdot)$.
- **Step S3.** For biometric authentication, *GWN* uses the following two fuzzy extractor functions:
  – *Gen*: It is a probabilistic generation function that takes as input the user personal biometrics Bio_i, and returns $\sigma_i \in \{0, 1\}^l$ that is the biometric key of length l bits and τ_i that is a public reproduction parameter.

– *Rep*: It is a deterministic function to be used during authentication. The input is the user biometrics, say Bio' and τ_i, provided the hamming distance between Bio' and the original previously entered biometrics Bio_i is less than t, where t is an error tolerance threshold value. The output is the original biometric key σ_i, that is, $\sigma_i = Rep(Bio'_i, \tau_i)$.

- **Step S4.** Finally, the system parameters $\{E_p(a, b), p, P, h(\cdot), Q_{GWN}, Gen(\cdot), Rep(\cdot), t\}$ are made public, whereas d_{GWN} is kept secret by GWN.

B. SENSING DEVICE REGISTRATION PHASE

All the sensing devices in IoT are registered offline by the GWN as follows.

- **Step SD1.** For each device SD_j, the GWN chooses a unique identity ID_j and a unique private key d_j, and calculates the corresponding public key $Q_j = d_j.P$. It further computes $RID_j = h(ID_j \| d_j)$.
- **Step SD2.** The GWN pre-loads $\{ID_j, d_j, RID_j\}$ in the memory of SD_j. Furthermore, the GWN stores $\{ID_j, RID_j, Q_j\}$ in its database, and then makes Q_j as public.

C. USER REGISTRATION PHASE

A user U_i registers with the GWN by executing the following steps:

- **Step R1.** U_i chooses a unique ID_i, a unique private key d_i and calculates the corresponding public key $Q_i = d_i.P$. U_i sends registration request message with $RID_i = h(ID_i \| d_i)$ to GWN via a secure channel.
- **Step R2.** GWN computes $R_i = h(RID_i \| d_{GWN})$, stores it on smart card SC_i and sends it to U_i via a secure channel.
- **Step R3.** U_i selects a password PW_i and imprints the biometrics template Bio_i at the sensor of a specific terminal. SC_i then computes the following:

$$\begin{aligned} Gen(Bio_i) &= (\sigma_i, \tau_i), \\ RPW_i &= h(PW_i \| d_i \| ID_i \| \sigma_i), \\ R_i^* &= R_i \oplus h(ID_i \| PW_i \| \sigma_i), \\ d_i^* &= d_i \oplus h(ID_i \| \sigma_i). \end{aligned}$$

- **Step R4.** U_i stores $\{d_i^*, RPW_i, Gen(\cdot), Rep(\cdot), \tau_i, h(\cdot), t\}$ and replaces R_i with R_i^* in SC_i. In addition, U_i also makes Q_i public.

The user registration phase has been summarized in Fig. 2.

User (U_i)	Gateway node (GWN)
Select ID_i, d_i. Compute $Q_i = d_i.P$ $RID_i = h(d_i \| ID_i)$. $\xrightarrow{\langle RID_i \rangle}$ (Secure channel)	
	Compute $R_i = h(RID_i \| d_{GWN})$. $\xleftarrow{\langle \text{Smart Card}\{R_i\} \rangle}$ (Secure channel)
Select PW_i. Imprint Bio_i. Compute $Gen(Bio_i) = (\sigma_i, \tau_i)$, $RPW_i = h(PW_i \| d_i \| ID_i \| \sigma_i)$, $R_i^* = R_i \oplus h(ID_i \| PW_i \| \sigma_i)$, $d_i^* = d_i \oplus h(ID_i \| \sigma_i)$. Insert $\{d_i^*, RPW_i, \tau_i, t, h(\cdot)$, $Gen(\cdot)$ and $Rep(\cdot)\}$ into smart card. Replace R_i with R_i^* in smart card.	

FIGURE 2. Summary of user registration phase.

D. LOGIN PHASE

U_i executes the following steps to login to the GWN:

- **Step L1.** After inserting SC_i, U_i enters his/her identity ID'_i and password PW'_i, and also imprints biometrics Bio'_i at the sensor of a specific terminal.
- **Step L2.** SC_i then computes $\sigma'_i = Rep(Bio'_i, \tau_i)$, $d'_i = d_i^* \oplus h(ID'_i \oplus \sigma'_i)$ and $RPW'_i = h(PW'_i \| ID'_i \| d'_i \| \sigma'_i)$, and checks if $RPW'_i = RPW_i$ holds.
- **Step L3.** If the above condition is verified successfully, U_i chooses a random secret number $a \in Z_p^*$, generates the current timestamp T_i and creates a login message with signature as follows:

$$\begin{aligned} A_i &= a.P = ((A_i)_x, (A_i)_y), \\ N_i &= a.Q_{GWN} = ((N_i)_x, (N_i)_y), \\ RID'_i &= h(d'_i \| ID'_i), \\ DID'_i &= RID'_i \oplus (N_i)_y, \\ DID'_j &= ID'_j \oplus (N_i)_y, \\ R'_i &= R_i^* \oplus h(ID'_i \| PW'_i \| \sigma'_i), \\ V_i &= h(ID_j \| T_i \| N_i \| R'_i), \\ r_i &= (N_i)_x, \\ s_i &= a^{-1}(V_i + r_i d'_i) \pmod p, \end{aligned}$$

where ID_j is the identity of the sensing device SD_j that U_i wants to communicate with. U_i finally sends $\{DID'_i, DID'_j, A_i, T_i, r_i, s_i\}$ to GWN as login message via a public channel.

E. AUTHENTICATION AND KEY AGREEMENT PHASE

In this phase, the GWN validates U_i and helps in establishing a session key between an accessed sensing device SD_j and a legal user U_i with the help of the following steps:

- **Step A1.** After receiving the login message from U_i at the time T'_i, the GWN first checks the validity of timestamp by the condition $T'_i - T_i \leq \Delta T$. If it is valid, the GWN then calculates $N_{GWN} = d_{GWN}.A_i = ((N_{GWN})_x, (N_{GWN})_y)$, $RID_i^* = DID'_i \oplus (N_{GWN})_y$, $ID_j^* = DID'_j \oplus (N_{GWN})_y$, $R_i = h(RID_i^* \| d_{GWN})$, $V_i^* = h(ID_j^* \| T_i \| N_{GWN} \| R_i)$.
  The GWN checks if ID_j^* is registered with it. If it is, then the GWN verifies U_i's signature as follows:

$$\begin{aligned} w_{GWN} &= s_i^{-1} \pmod p, \\ u_{GWN} &= V_i^* w_{GWN} \pmod p, \\ t_{GWN} &= r_i w_{GWN} \pmod p, \\ N_i^* &= ((N_i^*)_x, (N_i^*)_y) \\ &= (u_{GWN}.P + t_{GWN}.Q_i)d_{GWN}. \end{aligned}$$

User (U_i) $\{RPW_i, d_i^*, R_i^*, Gen(\cdot), Rep(\cdot), \tau_i, h(\cdot), t\}$	Gateway Node (GWN) $\{ID_j, RID_j, Q_j, d_{GWN}\}$	Sensing Device (SD_j) $\{ID_j, d_j, RID_j\}$
Enter ID_i' and PW_i'. Imprint Bio_i'. Compute $\sigma_i' = Rep(Bio_i', \tau_i)$, $d_i' = d_i^* \oplus h(ID_i' \parallel \sigma_i')$, $RPW_i' = h(PW_i' \parallel ID_i' \parallel d_i' \parallel \sigma_i')$. Check if $RPW_i' = RPW_i$? Choose random $a \in Z_p^*$. Generate timestamp T_i. Compute $A_i = a.P$, $N_i = a.Q_{GWN} = ((N_i)_x, (N_i)_y)$, $RID_i' = h(d_i' \parallel ID_i')$, $R_i' = R_i^* \oplus h(ID_i' \parallel PW_i \parallel \sigma_i')$, $DID_i' = RID_i' \oplus (N_i)_y$, $DID_j' = ID_j \oplus (N_i)_y$, $V_i = h(ID_j \parallel T_i \parallel N_i \parallel R_i')$, $r_i = (N_i)_x$, $s_i = a^{-1}(V_i + r_i d_i')$. $\langle DID_i', DID_j', A_i, T_i, r_i, s_i \rangle$ $\longrightarrow$	Check if $T_i' - T_i \leq \Delta T$? Compute $N_{GWN} = d_{GWN}.A_i$ $= ((N_{GWN})_x, (N_{GWN})_y)$, $RID_i^* = DID_i' \oplus (N_{GWN})_y$, $ID_j^* = DID_j' \oplus (N_{GWN})_y$. Check if $ID_j^* = ID_j$? If so, compute $R_i = h(RID_i^* \parallel d_{GWN})$, $V_i^* = h(ID_j^* \parallel T_i \parallel N_{GWN} \parallel R_i)$. Verify U_i's signature by computing $w_{GWN} = s_i^{-1} \pmod p$, $u_{GWN} = V_i^* w_{GWN} \pmod p$, $t_{GWN} = r_i w_{GWN} \pmod p$, $N_i^* = (u_{GWN}.P + t_{GWN}.Q_i) d_{GWN}$ $= ((N_i^*)_x, (N_i^*)_y)$. Check if $(r_i^* = (N_i^*)_x) = ((N_i)_x = r_i)$?	
	Choose random $c \in Z_p^*$. Generate timestamp T_{GWN}. Compute $C_{GWN} = c.P$ $= ((C_{GWN})_x, (C_{GWN})_y)$, $V_{GWN} = h(R_i \parallel T_i)$ $\oplus h(A_i \parallel RID_j \parallel T_{GWN} \parallel T_i)$, $r_{GWN} = (C_{GWN})_x$, $s_{GWN} = c^{-1}(h(R_i \parallel T_i)+$ $r_{GWN} d_{GWN}) \pmod p$. $\langle V_{GWN}, T_{GWN}, T_i, A_i, C_{GWN}, s_{GWN} \rangle$ $\longrightarrow$	Check if $T_{GWN}' - T_{GWN} \leq \Delta T$? Compute $h(R_i \parallel T_i) = V_{GWN} \oplus$ $h(A_i \Vert RID_j \Vert T_{GWN} \Vert T_i)$. Verify GWN's signature by computing $w_{SD_j} = s_{GWN}^{-1} \pmod p$, $u_{SD_j} = h(R_i \parallel T_i) w_{SD_j} \pmod p$, $r_{GWN} = (C_{GWN})_x$, $t_{SD_j} = r_{GWN} w_{SD_j} \pmod p$, $C_{GWN}^* = u_{SD_j}.P + t_{SD_j}.Q_{GWN}$ $= ((C_{GWN}^*)_x, (C_{GWN}^*)_y)$. Check if $(r_{GWN}^* = (C_{GWN}^*)_x) =$ $((C_{GWN})_x = r_{GWN})$?
Check if $T_j' - T_j \leq \Delta T$? Compute $k_{ij}' = a.B_{SD_j} = a.(b.P)$, $sk_{ij}' = h(ID_j \parallel h(R_i^* \parallel T_i) \parallel k_{ij}' \parallel T_i \parallel T_j)$, Verify SD_j's signature by computing $w_i = s_{SD_j}^{-1} \pmod p$, $u_i = h(sk_{ij}') w_i \pmod p$, $r_{SD_j} = (B_{SD_j})_x$, $t_i = r_{SD_j} w_i \pmod p$, $B_{SD_j}^* = u_i.P + t_i.Q_j = ((B_{SD_j}^*)_x, (B_{SD_j}^*)_y)$. Check if $(r_{SD_j}^* = (B_{SD_j}^*)_x) = ((B_{SD_j})_x = r_{SD_j})$? Store the session key sk_{ij}' shared with SD_j.		Choose random $b \in Z_p^*$. Generate timestamp T_j. Compute $k_{ij} = b.A_i = b.(a.P)$, $sk_{ij} = h(ID_j \parallel h(R_i \parallel T_i) \parallel k_{ij} \parallel T_i \parallel T_j)$, $B_{SD_j} = b.P = ((B_{SD_j})_x, (B_{SD_j})_y)$, $r_{SD_j} = (B_{SD_j})_x$, $s_{SD_j} = b^{-1}(h(sk_{ij}) + r_{SD_j} d_j) \pmod p$. $\langle B_{SD_j}, s_{SD_j}, T_j \rangle$ $\longleftarrow$ (to U_i) Store the session key sk_{ij} shared with U_i.

FIGURE 3. Summary of login and authentication phases.

Note that $(u_{GWN}.P + t_{GWN}.Q_i)\ d_{GWN} = (((V_i^*P)/s_i) + (((r_i d_i).P)/s_i))\ d_{GWN} = (1/s_i)\ (V_i^* + r_i d_i)\ d_{GWN}.P = (1/s_i)(a s_i)\ d_{GWN}.P = a.Q_{GWN} = N_i = ((N_i)_x, (N_i)_y)$. GWN checks if $r_i^* = (N_i^*)_x = (N_i)_x = r_i$ as explained above to verify U_i's signature.

- **Step A2.** After successful signature verification, GWN chooses a random secret number $c \in Z_p^*$, generates its current timestamp T_{GWN} and computes the following message with signature:

$$\begin{aligned} C_{GWN} &= c.P = ((C_{GWN})_x, (C_{GWN})_y), \\ V_{GWN} &= h(R_i \parallel T_i) \oplus h(A_i \parallel RID_j \parallel T_{GWN} \parallel T_i), \\ r_{GWN} &= (C_{GWN})_x, \\ s_{GWN} &= c^{-1}(h(R_i \parallel T_i) + r_{GWN} d_{GWN}) \pmod p. \end{aligned}$$

GWN then sends authentication request message $\{V_{GWN}, T_{GWN}, T_i, A_i, C_{GWN}, s_{GWN}\}$ to SD_j via a public channel.

- **Step A3.** If SD_j receives the message at time T_{GWN}', it verifies the timeliness of T_{GWN} by $T_{GWN}' - T_{GWN} \leq \Delta T$. If it is valid, SD_j then computes

$$\begin{aligned} h(R_i \parallel T_i) &= V_{GWN} \oplus h(A_i \parallel RID_j \parallel T_{GWN} \parallel T_i), \\ w_{SD_j} &= s_{GWN}^{-1} \pmod p, \\ u_{SD_j} &= h(R_i \parallel T_i) w_{SD_j} \pmod p, \\ r_{GWN} &= (C_{GWN})_x, \\ t_{SD_j} &= r_{GWN} w_{SD_j} \pmod p, \\ C_{GWN}^* &= u_{SD_j}.P + t_{SD_j}.Q_{GWN} \\ &= ((C_{GWN}^*)_x, (C_{GWN}^*)_y). \end{aligned}$$

Note that $u_{SD_j}.P + t_{SD_j}.Q_{GWN} = h(R_i \parallel T_i)w_{SD_j}.P + r_{GWN}w_{SD_j}(d_{GWN}.P) = w_{SD_j}(h(R_i \parallel T_i) + r_{GWN}d_{GWN}).P = (1/s_{GWN})(cs_{GWN}).P = c.P = C_{GWN} = ((C_{GWN})_x, (C_{GWN})_y)$.
SD_j then checks if $r^*_{GWN} = (C^*_{GWN})_x = (C_{GWN})_x = r_{GWN}$ as shown above to verify GWN's signature. After successful signature verification, SD_j chooses a random number $b \in Z^*_p$, generates its current timestamp T_j, and computes the session key with signature as follows:

$$\begin{aligned} k_{ij} &= b.A_i = b.(a.P), \\ sk_{ij} &= h(ID_j \parallel h(R_i \parallel T_i) \parallel k_{ij} \parallel T_i \parallel T_j), \\ B_{SD_j} &= b.P = ((B_{SD_j})_x, (B_{SD_j})_y), \\ r_{SD_j} &= (B_{SD_j})_x, \\ s_{SD_j} &= b^{-1}(h(sk_{ij}) + r_{SD_j}d_j) \pmod p. \end{aligned}$$

SD_j sends authentication reply message with $\{B_{SD_j}, s_{SD_j}, T_j\}$ to U_i via open channel.

- **Step A4.** U_i receives SD_j's authentication message at time T'_j and verifies if $T'_j - T_j \leq \Delta T$. If the validity of timestamp passes, U_i verifies SD_j's signature and computes the session key as follows:

$$\begin{aligned} k'_{ij} &= a.B_{SD_j} = a.(b.P) = k_{ij}, \\ sk'_{ij} &= h(ID_j \parallel h(R^*_i \parallel T_i) \parallel k'_{ij} \parallel T_i \parallel T_j), \\ w_i &= s^{-1}_{SD_j} \pmod p, \\ u_i &= h(sk'_{ij})w_i \pmod p, \\ t_i &= r_{SD_j}w_i \pmod p, \\ B^*_{SD_j} &= u_i.P + t_i.Q_j \\ &= ((B^*_{SD_j})_x, (B^*_{SD_j})_y). \end{aligned}$$

Note that $u_i.P + t_i.Q_j = (h(sk'_{ij})w_i).P + (r_{SD_j}w_id_j).P = w_i(h(sk'_{ij}) + r_{SD_j}d_j).P = (1/s_{SD_j})(b.s_{SD_j}).P = b.P = ((B_{SD_j})_x, (B_{SD_j})_y)$.
U_i checks if $r^*_{SD_j} = (B^*_{SD_j})_x = (B_{SD_j})_x = r_{SD_j}$ as noted above, and establishes secure communication with SD_j using the session key sk_{ij}.

The summary of login and authentication phases is provided in Fig. 3.

F. PASSWORD AND BIOMETRIC UPDATE PHASE

U_i executes this phase internally without involving the GWN to reduce overhead as follows:

- **Step PB1.** U_i enters his/her identity ID_i, current password PW^{old}_i and imprints current biometrics Bio^{old}_i at the sensor of a specific terminal. SC_i then computes

$$\begin{aligned} \sigma^{old}_i &= Rep(Bio^{old}_i, \tau_i), \\ d'_i &= d^*_i \oplus h(ID_i \parallel \sigma^{old}_i), \\ R'_i &= R^*_i \oplus h(ID_i || PW^{old}_i || \sigma^{old}_i), \\ RPW^{old}_i &= h(PW^{old}_i \parallel d'_i \parallel ID_i \parallel \sigma^{old}_i). \end{aligned}$$

SC_i checks if $RPW^{old}_i = RPW_i$ and the request is terminated if the verification is not successful.

- **Step PB2.** U_i then enters new password PW^{new}_i and imprints new biometric Bio^{new}_i. SC_i computes the following:

$$\begin{aligned} Gen(Bio^{new}_i) &= (\sigma^{new}_i, \tau^{new}_i), \\ RPW^{new}_i &= h(PW^{new}_i \parallel d'_i \parallel ID_i \parallel \sigma^{new}_i), \\ (d^*_i)^{new} &= d'_i \oplus h(ID_i \parallel \sigma^{new}_i), \\ (R^*_i)^{new} &= R'_i \oplus h(ID_i || PW^{new}_i || \sigma^{new}_i). \end{aligned}$$

- **Step PB3.** RPW_i, d^*_i, R^*_i and τ_i on SC_i are replaced with RPW^{new}_i, $(d^*_i)^{new}$, $(R^*_i)^{new}$ and τ^{new}_i, respectively.

This phase has been summarized in Fig. 4.

User (U_i)	Smart card (SC_i)
Enter $ID_i, PW^{old}_i, Bio^{old}_i$. $\xrightarrow{\{ID_i, PW^{old}_i, Bio^{old}_i\}}$	
	Compute $\sigma^{old}_i = Rep(Bio^{old}_i, \tau_i)$, $d'_i = d^*_i \oplus h(ID_i \parallel \sigma^{old}_i)$, $R'_i = R^*_i \oplus h(ID_i \parallel PW^{old}_i \parallel \sigma^{old}_i)$, $RPW^{old}_i = h(PW^{old}_i \parallel d'_i \parallel ID_i \parallel \sigma^{old}_i)$. If $RPW^{old}_i = RPW_i$ does not hold, terminate. $\xleftarrow{\{\text{Permit user to change password/biometric}\}}$
Enter PW^{new}_i, Bio^{new}_i. $\xrightarrow{\{PW^{new}_i, Bio^{new}_i\}}$	
	Compute $Gen(Bio^{new}_i) = (\sigma^{new}_i, \tau^{new}_i)$, $RPW^{new}_i = h(PW^{new}_i \parallel d'_i \parallel ID_i \parallel \sigma^{new}_i)$, $(d^*_i)^{new} = d'_i \oplus h(ID_i \parallel \sigma^{new}_i)$, $(R^*_i)^{new} = R'_i \oplus h(ID_i \parallel PW^{new}_i \parallel \sigma^{new}_i)$. Replace the old values RPW_i, d^*_i, R^*_i and τ_i with new ones RPW^{new}_i, $(d^*_i)^{new}$, $(R^*_i)^{new}$, and τ^{new}_i, respectively.

FIGURE 4. Summary of password and biometric update phase.

G. SMART CARD REVOCATION PHASE

If the smart card SC_i of a legitimate user U_i is lost, the following steps can be executed for requesting a new one:

- **Step RV1.** U_i creates a registration request message with the same ID_i and new private key d^{new}_i as $RID^{new}_i = h(d^{new}_i \parallel ID_i)$ and sends it to the GWN via a secure channel.
- **Step RV2.** GWN computes $R^{new}_i = h(RID^{new}_i \parallel d_{GWN})$ and sends SC^{new}_i to U_i with R^{new}_i stored in it via a secure channel.
- **Step RV3.** U_i then uses the current PW_i and Bio_i to compute the following:

$$\begin{aligned} Gen(Bio_i) &= (\sigma_i, \tau_i), \\ Q^{new}_i &= d^{new}_i.P, \\ RPW^{new}_i &= h(PW_i \parallel d^{new}_i \parallel ID_i \parallel \sigma_i), \\ (R^*_i)^{new} &= R^{new}_i \oplus h(ID_i \parallel PW_i \parallel \sigma_i), \\ (d^*_i)^{new} &= d^{new}_i \oplus h(ID_i \parallel \sigma_i). \end{aligned}$$

- **Step RV4.** SC^{new}_i is personalized with the values $\{RPW^{new}_i$, $(R^*_i)^{new}$, $(d^*_i)^{new}$, τ_i, t, $h(\cdot)$, $Gen(\cdot)$ and $Rep(\cdot)\}$.

This phase has been summarized in Fig. 5.

User (U_i)	Gateway node (GWN)
Select d_i^{new}. Enter current ID_i. Compute $Q_i^{new} = d_i^{new}.P$ $RID_i^{new} = h(d_i^{new} \parallel ID_i)$. $\xrightarrow{\langle RID_i^{new} \rangle}$	
	Compute $R_i^{new} = h(RID_i^{new} \parallel d_{GWN})$. $\xleftarrow{\langle \text{Smart Card}\{R_i^{new}\} \rangle}$
Use current PW_i and Bio_i. Enter PW_i and imprint Bio_i. Compute $Gen(Bio_i) = (\sigma_i, \tau_i)$, $RPW_i^{new} = h(PW_i \parallel d_i^{new} \parallel ID_i \parallel \sigma_i)$, $Q_i^{new} = d_i^{new}.P$, $(R_i^*)^{new} = R_i^{new} \oplus h(ID_i \parallel PW_i \parallel \sigma_i)$, $(d_i^*)^{new} = d_i^{new} \oplus h(ID_i \parallel \sigma_i)$. Insert $\{(d_i^*)^{new}, RPW_i^{new}, \tau_i, t, h(\cdot),$ $(R_i^*)^{new}\ Gen(\cdot)$ and $Rep(\cdot)\}$ into smart card. Make Q_i^{new} public.	

FIGURE 5. Summary of smart card revocation phase.

H. DYNAMIC SENSING DEVICE ADDITION PHASE

Dynamic sensing device addition is necessary as some devices may be physically compromised by an attacker and we need to deploy some new devices in the network. Suppose a new sensing device SD_j^{new} is to be deployed in the network. The GWN then performs the following steps offline:

- **Step DSD1.** The GWN chooses a unique identity ID_j^{new} and a unique private key d_j^{new}, and calculates the corresponding public key $Q_j^{new} = d_j^{new}.P$. It further computes $RID_j^{new} = h(ID_j^{new} \parallel d_j^{new})$.
- **Step DSD2.** The GWN pre-loads RID_j^{new} in the memory of SD_j^{new}. In addition, the GWN stores $\{ID_j^{new}, RID_j^{new}, Q_j^{new}\}$ in its database, and also makes Q_j^{new} public.

After the deployment of SD_j^{new}, the GWN informs the users in the network so that they can access SD_j^{new} using the login and authentication & key agreement phases described in Sections V-D and V-E, respectively.

VI. SECURITY ANALYSIS OF THE PROPOSED SCHEME

In this section, we first prove that the proposed scheme provides secure mutual authentication between a user U_i and a sensing device SD_j with the help of the widely-accepted BAN logic. Furthermore, we show that the proposed scheme is secure against various known attacks informally. In addition, the formal security verification using the broadly-accepted AVISPA tool ensures that the scheme is also secure against replay and man-in-the-middle attacks.

A. MUTUAL AUTHENTICATION USING BAN LOGIC

To prove that a user U_i and a sensing device SD_j mutually authenticate each other through fresh and trustworthy information, the BAN logic is being used. This is achieved by verifying the message's origin, the origin's freshness and trustworthiness. The following notations are used in the BAN logic:

- $A \mid\equiv X$: A believes the statement X.
- $A \lhd X$: A sees X, i.e. A has received a message containing X.
- $A \mid\sim X$: A once said X i.e $A \mid\equiv X$ when A sent it.
- $A \mid \Longrightarrow X$: A has authority or jurisdiction over X.
- $\#(X)$: X is a fresh message.
- $A \overset{K}{\longleftrightarrow} B$: K is shared secret key between A and B.
- X_K: X is encrypted with key K.
- $< X >_Y$: formula X is combined with formula Y.
- $(X)_K$: X is hashed with key K.
- (X, Y): X or Y is one part of formula (X, Y).

The logical postulates in the BAN logic are described using the below mentioned rules:

Rule 1 (Message Meaning Rule (MMR)): P believes Q once said X if P sees a message X encrypted with K and P believes K is a shared secret between P and Q.

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

$$\frac{P \mid\equiv P \overset{Y}{\longleftrightarrow} Q, P \lhd \langle X \rangle_Y}{P \mid\equiv Q \mid\sim X}.$$

Rule 2 (Nonce Verification Rule (NVR)): P believes Q believes X if P believes Q once said X and P believes X is fresh.

$$\frac{P \mid\equiv \#\{X\} P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}.$$

Rule 3 (Jurisdiction Rule (JR)): P believes X if P believes that Q believes X and P believes Q has jurisdiction over X.

$$\frac{P \mid\equiv Q \mid\equiv X, P \mid\equiv Q \mapsto X}{P \mid\equiv X}.$$

Rule 4 (Freshness Rule (FR)): The entire formula is believed to be fresh if a part of the formula is believed to be fresh, .

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

Rule 5 (Belief Rule (BR)): P believes Q believes part of the formula if P believes Q believes a formula,

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

P believes combined formula (X, Y) if P believes X and P also believes Y.

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

Theorem 1: The proposed scheme provides the secure mutual authentication between a user U_i and a sensing device SD_j.

Proof: The login and authentication phases involve exchanging of messages whose generic form can be expressed as follows:

Message 1 ($GWN \to SD_j$): $(h(R_i \parallel T_i) \oplus h(a.P \parallel RID_j \parallel T_{GWN} \parallel T_i), c.P, c^{-1}(h(R_i \parallel T_i) + d_{GWN}\ (c.P)_x), a.P, T_{GWN}, T_i)$.

Message 2 ($SD_j \rightarrow U_i$): ($b.P$, $b^{-1}(h(h(ID_j \parallel h(R_i \parallel T_i) \parallel b.a.P \parallel T_i \parallel T_j)) + d_{SD_j}\ (b.P)_x)$, T_j). *Idealized form:* The ideal forms of the above messages can be expressed as follows:

Message 1 ($GWN \rightarrow SD_j$): $\langle <\ ((R_i \parallel T_i), (a.P \parallel RID_j \parallel T_i \parallel T_{GWN})), c.P, ((R_i \parallel T_i), d_{GWN}(c.P)), T_{GWN}, T_i > \rangle_{GWN \overset{(a.P)}{\longleftrightarrow} SD_j}$.

Message 2 ($SD_j \rightarrow U_i$): $\langle <\ (U_i \overset{sk_{ij}}{\longleftrightarrow} SD_j), d_{SD_j} (b.P)_x)), T_j > \rangle_{U_i \overset{b.P}{\longleftrightarrow} SD_j}$. *Goal:* The goals to be proven are the following:

G1: $U_i \mid\equiv U_i \overset{sk_{ij}}{\longleftrightarrow} SD_j$,

G2: $SD_j \mid\equiv U_i \overset{sk_{ij}}{\longleftrightarrow} SD_j$,

using the assumptions mentioned below:

A1. $U_i \mid\equiv \#(T_i)$, $U_i \mid\equiv \#(T_j)$;

A2. $GWN \mid\equiv \#(T_i)$, $U_i \mid\equiv \#(T_{GWN})$;

A3. $SD_j \mid\equiv \#(T_i)$, $SD_j \mid\equiv \#(T_{GWN})$, $SD_j \mid\equiv \#(T_j)$;

A4. $GWN \mid\equiv (GWN \overset{a.P}{\longleftrightarrow} SD_j)$;

A5. $SD_j \mid\equiv (GWN \overset{a.P}{\longleftrightarrow} SD_j)$;

A6. $SD_j \mid\equiv GWN \Rightarrow GWN \mid\sim X$;

A7. $U_i \mid\equiv (U_i \overset{b.P}{\longleftrightarrow} SD_j)$;

A8. $SD_j \mid\equiv (U_i \overset{b.P}{\longleftrightarrow} SD_j)$;

A9. $U_i \mid\equiv SD_j \Rightarrow (U_i \overset{sk_{ij}}{\longleftrightarrow} SD_j)$.

The mutual authentication between U_i and SD_j is as follows:

S1. From message 1, we get,
$SD_j \triangleleft \langle <\ ((R_i \parallel T_i), (a.P \parallel RID_j \parallel T_i \parallel T_{GWN})), c.P, ((R_i \parallel T_i), d_{GWN}\ (c.P)), T_{GWN}, T_i > \rangle_{GWN \overset{(a.P)}{\longleftrightarrow} SD_j}$.

S2. Using S1, A5 and MMR, we obtain,
$SD_j \mid\equiv GWN \mid\sim \langle < ((R_i \parallel T_i), (a.P \parallel RID_j \parallel T_i \parallel T_{GWN})), c.P, ((R_i \parallel T_i), d_{GWN}(c.P)), T_{GWN}, T_i > \rangle$.

S3. Using S2, A3, FR and NVR, it follows that
$SD_j \mid\equiv GWN \mid\equiv \langle < ((R_i \parallel T_i), (a.P \parallel RID_j \parallel T_i \parallel T_{GWN})), c.P, ((R_i \parallel T_i), d_{GWN}\ (c.P)), T_{GWN}, T_i > \rangle$.

S4. Using A6, S3, JR and BR, we get $SD_j \mid\equiv (R_i \parallel T_i)$.

S5. Using S4 and BR, we get,
$SD_j \mid\equiv U_i \overset{sk_{ij}}{\longleftrightarrow} SD_j$. **(Goal G2)**

S6. From message 2, we get,
$U_i \triangleleft \langle < ((U_i \overset{sk_{ij}}{\longleftrightarrow} SD_j), d_{SD_j}(b.P)_x)), T_j > \rangle_{U_i \overset{b.P}{\longleftrightarrow} SD_j}$.

S7. Using S6, A7 and MMR, we get,
$U_i \mid\equiv SD_j \mid\sim \langle < ((U_i \overset{sk_{ij}}{\longleftrightarrow} SD_j), d_{SD_j}(b.P)_x)), T_j > \rangle$.

S8. Using S7, A1, FR, NVR and BR, we get,
$U_i \mid\equiv SD_j \mid\equiv U_i \overset{sk_{ij}}{\longleftrightarrow} SD_j$.

S9. Using S8, A9 and JR, we get,
$U_i \mid\equiv U_i \overset{sk_{ij}}{\longleftrightarrow} SD_j$. **(Goal G1)**

The goals G1 and G2 clearly show that U_i and SD_j mutually authenticate each other with help from the GWN. □

B. DISCUSSION ON OTHER ATTACKS

An informal analysis in the following sections shows that the proposed scheme is secure against various well-known attacks, and it also provides the required functionality features.

1) PRIVILEGED-INSIDER ATTACK

A privileged user at the GWN, who may be an adversary $\mathcal{A}$, can obtain RID_i, which is the user U_i's registration information during the user registration phase. Suppose the smart card SC_i of U_i is lost or stolen by $\mathcal{A}$ after the registration process is completed. Even by retrieving all stored information from SC_i using the power analysis attacks [4], [5], such as $\{RPW_i, \tau_i, R_i^*, d_i^*\}$, neither ID_i nor PW_i can be guessed by $\mathcal{A}$. This is because U_i's private key d_i is used in masking ID_i which is not stored directly in SC_i. Also, RPW_i and d_i^* stored in SC_i are protected through the one-way hash function $h(\cdot)$. To correctly guess ID_i as well as PW_i, $\mathcal{A}$ also needs to know the biometric key σ_i and the private key d_i. To derive d_i, $\mathcal{A}$ needs to know both ID_i and σ_i as $d_i^* = d_i \oplus h(ID_i||\sigma_i)$. Thus, the proposed scheme is secure against this attack.

2) USER IMPERSONATION ATTACK

Assume an intruder $\mathcal{I}$ tries to create a valid login request message by impersonating U_i after obtaining U_i's login request message $\{DID_i', DID_j', A_i, r_i, s_i, T_i\}$. For this, $\mathcal{I}$ can select a random number $a' \in Z_p^*$ and attempt to compute $A_i' = a'.P$, $N_i' = a'.Q_{GWN} = (N_i'^x, N_i'^y)$, $DID_i' = RID_i \oplus N_i'^y$, $DID_j' = ID_j \oplus N_i'^y$ and $V_i' = h(T_i' \parallel N_i' \parallel R_i')$. Here, $\mathcal{I}$ needs to know ID_j of the sensing device SD_j that U_i is attempting to communicate with, RID_i of U_i and private key d_{GWN} of GWN to compute $R_i^* = h(RID_i \parallel d_{GWN})$ to be able to successfully compute V_i'. Hence, recreating login request by eavesdropping is impossible and it makes our scheme secure against this attack.

3) OFFLINE PASSWORD GUESSING ATTACK

Suppose an adversary $\mathcal{A}$ knows all information in smart card SC_i of U_i, that is, $\{RPW_i, R_i^*, d_i^*, \tau_i, h(\cdot)\}$ using the power analysis attacks [4], [5]. $\mathcal{A}$ cannot derive U_i's password PW_i because of hash function $h(\cdot)$'s one-way property which protects ID_i, d_i and σ_i from $\mathcal{A}$. Therefore, the proposed scheme is secure against such an attack.

4) STOLEN SMART CARD ATTACK

A lost/stolen smart card SC_i of U_i reveals all the stored information $\{RPW_i, R_i^*, d_i^*, \tau_i, h(\cdot)\}$ to an adversary $\mathcal{A}$. However, U_i's secret credentials are not revealed as $h(\cdot)$ and U_i's private key d_i protect the values ID_i, σ_i and PW_i. Thus, the proposed scheme is secure against this attack.

5) DENIAL-OF-SERVICE ATTACK

Even if a legal user U_i enters incorrect ID_i and/or PW_i during login phase, it is locally detected through the verification $RPW_i' = RPW_i$ (Step L2 in Section V-D). The login request to the GWN is sent only after successful verification. Therefore, the proposed scheme is safe from this attack.

6) REPLAY ATTACK

As the current time stamps of all involved entities GWN, U_i and SD_j are used in all communicated messages with a sufficiently small acceptable delay interval, ΔT, an adversary $\mathcal{A}$ cannot replay login or authentication messages obtained by eavesdropping. As a result, the replay attack is prevented in the proposed scheme.

7) MAN-IN-THE-MIDDLE ATTACK

Suppose an adversary $\mathcal{A}$ intercepts the login request message $\{DID'_i, DID'_j, A_i, T_i, r_i, s_i\}$ and tries to modify this message to another valid login request message. For this purpose, $\mathcal{A}$ can select a random number $a^* \in Z_p$ and generate a current timestamp T_i^*. Then, $\mathcal{A}$ can calculate $A_i^* = a^*.P$, $N_i^* = a^*.Q_{GWN} = ((N_i^*)_x, (N_i^*)_y)$ and $r_i^* = (N_i^*)_x$. However, without ID_i, PW_i, d_i and σ_i, $\mathcal{A}$ can not compute $RID_i = h(d_i \parallel ID_i)$ and $R_i = h(RID_i \parallel d_{GWN})$, where d_{GWN} is the private key of the GWN. Furthermore, without the private key d_i of U_i, R_i and ID_j, it is a difficult task for $\mathcal{A}$ to calculate the modified $V_i^* = h(ID_j \parallel T_i^* \parallel N_i^* \parallel R_i)$ and the signature $s_i^* = (a^*)^{-1}(V_i^* + r_i^* d_i)$. Hence, $\mathcal{A}$ can not create a valid login request message, say $\{DID_i^*, DID_j^*, A_i^*, T_i^*, r_i^*, s_i^*\}$. In a similar way, $\mathcal{A}$ can not also create other messages during the authentication and key establishment phase. Therefore, the proposed scheme is secure against man-in-the-middle attack.

8) RESILIENCE AGAINST SENSING DEVICE ATTACK

Similar to wireless sensor network user authentication [30], [31], we also measure the resilience against sensing device capture attack of a user authentication scheme in IoT environment. Suppose c sensing devices are physically captured by an adversary $\mathcal{A}$. We then estimate the fraction of total secure communications that are compromised by a capture of c sensing devices *not including* the communication in which the compromised sensing devices are directly involved. For example, one can find out the probability that $\mathcal{A}$ can decrypt the secure communication between a user and a non-compromised sensing device when c sensing devices are already compromised. If this probability is denoted by $P_e(c)$ and $P_e(c) = 0$, a user authentication scheme is called unconditionally secure against sensing device capture attack.

Let $\mathcal{A}$ capture a sensing device SD_j. $\mathcal{A}$ can then extract the information $\{ID_j, d_j, RID_j\}$ from its memory using power analysis attacks [4], [5]. Note that all these ID_j, d_j and RID_j are distinct for all the sensing devices in IoT, and these are generated by the GWN. Hence, by capturing SD_j, $\mathcal{A}$ can only compromise the session key between that a user and SD_j. However, all other session keys between that user and other non-compromised sensing devices are not compromised by $\mathcal{A}$. As a result, compromise of a sensing device does not lead to compromise of the secure communications among a user and other sensing devices, and therefore, the proposed scheme is unconditionally secure against sensing device capture attack.

9) ANONYMITY AND UNTRACEABILITY

Assume that an adversary $\mathcal{A}$ intercepts $Msg_1 = \{DID'_i, DID'_j, A_i, T_i, r_i, s_i\}$, $Msg_2 = \{V_{GWN}, T_{GWN}, T_i, A_i, C_{GWN}, s_{GWN}\}$ and $Msg_3 = \{B_{SD_j}, s_{SD_j}, T_j\}$ during the login & authentication phases. Due to random number a and current timestamp T_i, each of DID'_i, DID'_j, A_i, T_i, r_i and s_i are dynamic and "unique" in Msg_1 for each session. Similarly, due to random numbers and current timestamps used, Msg_2 and Msg_3 are also dynamic and "unique" for each session. Furthermore, none of these messages directly includes the identities ID_i and ID_j in the plaintext transmission over insecure channels. Hence, the proposed scheme preserves both anonymity and untraceability properties.

C. FORMAL SECURITY VERIFICATION USING AVISPA TOOL

In this section, we simulate the proposed scheme using broadly-accepted AVISPA tool [32]. We provide the implementation details of our scheme in high-level protocol specification language (HLPSL) [33] and then the simulation results to show our scheme is secure against replay and man-in-the-middle attacks.

1) HLPSL IMPLEMENTATION

The HLPSL implementation for registration, login and authentication/key agreement phases involves three basic roles: *user* (shown in Fig. 6) for a user U_i, *gwn* (shown in Fig. 7) for the gateway node GWN and *sensingdevice* (shown in Fig. 8) for a sensing device SD_j. The implementation also requires defining the necessary roles for the session, and goal and environment (shown in Fig. 9.

After receiving the start signal to begin the communication, U_i alters the value of variable *State* to 1 from 0. During registration, U_i sends a registration request message $\langle RID_i \rangle$ via a secure channel to GWN. GWN changes its state to 2 from 0 and replies via a secure channel with a smart card with $\{R_i\}$ stored on it. U_i then alters its state to 2 from 1. The login phase is then initiated by U_i by sending a login request message $\langle DID'_i, DID'_j, A_i, T_i, r_i, s_i \rangle$ to GWN via an open channel. Upon receiving the message, GWN alters its state to 4 from 2. GWN then forwards an authentication request message with $\langle V_{GWN}, T_{GWN}, T_i, A_i, C_{GWN}, s_{GWN} \rangle$ to SD_j over a public channel to initiate the authentication and key establishment phase. Once it receives the message, SD_j changes its state to 3 from 0 and responds by sending an authentication reply message with $\langle B_{SD_j}, s_{SD_j}, T_j \rangle$ to U_i over a public channel.

In the role for SD_j, the witness declaration witness(SDj, Ui, sdj_ui_b, B') means that $b \in Z_p^*$ has been chosen freshly for U_i by SD_j. The request declaration request(SDj, Ui, sdj_ui_b, B') in role for U_i indicates that U_i has accepted the value b generated for it by SD_j. The secret declaration secret({Dgwn}, sec2, {GWN}) in the role for GWN indicates that GWN keeps its private key d_{GWN} as secret. The protocol id *sec*2 characterizes this declaration. Similarly, all other witness, request and secret declarations have been defined. In our

```
role user (Ui, GWN, SDj : agent, SKugwn : symmetric_key,
        Snd, Rcv: channel(dy))
played_by Ui
def=
local State : nat, H, F, G: hash_func,
    P, Di, Qi, IDi, RIDi, Dgwn, A, B, Ai : text,
    DIDi, DIDj, IDj, Ni, Ti, Tj, Dj, Vi, Si : text
const sec1, ui_gwn_a, ui_gwn_ti, sdj_ui_b,
    sdj_ui_tj : protocol_id
init State := 0
transition
% User Registration Phase
1. State = 0 /\ Rcv(start) =|>
State' := 1 /\ Qi' := F(Di.P) /\ RIDi' := H(IDi.Di)
% Send registration request to the GWN securely
        /\ Snd({RIDi'}_SKugwn)
        /\ secret(Di, sec1, {Ui})
% Receive smart card from the GWN securely
2. State = 1 /\ Rcv({H(H(IDi.Di).Dgwn)}_SKugwn) =|>
% Login and Authentication Phases
State' := 2 /\ A' := new() /\ Ti' := new() /\ Ai' := F(A'.P)
        /\ Ni' := F(A'.F(Dgwn.P))
        /\ DIDi' := xor(H(IDi.Di), Ni')
        /\ DIDj' := xor(IDj,Ni')
        /\ Vi' := H(Ti'.Ni'.H(H(IDi.Di).Dgwn))
        /\ Si' := G(A'.Vi'.Ni'.Di)
% Send login request message to the GWN via open channel
        /\ Snd(DIDi'.DIDj'.Ai'.Ti'.Ni'.Si')
% Ui has freshly generated the values a and Ti for GWN
      /\ witness (Ui, GWN, ui_gwn_a, A')
      /\ witness (Ui, GWN, ui_gwn_ti, Ti')
% Receive authentication reply from SDj via open channel
3. State = 2 /\ Rcv(F(B'.P).G(B'.H(H(IDj.H(H(H(IDi.Di).Dgwn).Ti').
              F(B'.F(A'.P)).Ti'.Tj').F(B'.P).Dj)).Tj')  =|>
% Ui's acceptance of the values b and Tj generated for Ui by SDj
State' := 4 /\ request(SDj, Ui, sdj_ui_b, B')
        /\ request(SDj, Ui, sdj_ui_tj, Tj')
end role
```

FIGURE 6. Role specification in HLPSL for the user U_i.

implementation, three secrecy goals and six authentication goals are required.

The intruder (i) has also been shown as one of the participants through a concrete session in the protocol execution.

2) ANALYSIS OF RESULTS

We have chosen the broadly-used On-the-fly Model-Checker (OFMC) and Constraint Logic based Attack Searcher (CL-AtSe) backends for the execution test to find whether there are any attacks on the proposed scheme [32]. To check for the possibility of a replay attack, these back-ends verify if the specified protocol can be executed by the legitimate agents by searching for a passive intruder. The back-ends provide the intruder (i) with information about a few normal sessions between the legitimate agents. To check the Dolev-Yao model, the back-ends also verify if there is any possibility of a man-in-the-middle attack by the intruder.

All public parameters are known to the intruder. We have simulated the proposed scheme using SPAN, the Security Protocol ANimator for AVISPA [34], for both OFMC and CL-AtSe backends. The simulation results of the analysis using these backends shown in Fig. 10 ensure that the proposed scheme is safe against replay and man-in-the-middle attacks. The output in Fig. 10 has the following sections:

SUMMARY: This either indicates that the scheme has been found to be safe or unsafe or that the analysis has been inconclusive.

```
role gwn (Ui, GWN, SDj : agent, SKugwn : symmetric_key,
        Snd, Rcv: channel(dy))
played_by GWN
def=
local State : nat, H, F, G: hash_func,
    P, Di, IDi, Ri, Dgwn, A, IDj, Dj, Ti, C, Tgwn : text,
    Cgwn, Vgwn, Sgwn : text
const sec2, ui_gwn_a, ui_gwn_ti, gwn_sdj_c, gwn_sdj_tgwn : protocol_id
init State := 0
transition
% User Registration Phase
1. State = 0 /\ Rcv({H(IDi.Di)}_SKugwn) =|>
State' := 2  /\ Ri':= H(H(IDi.Di).Dgwn)
% Send smart card to Ui securely
          /\ Snd({Ri'}_SKugwn)
          /\ secret(Dgwn, sec2, {GWN})
% Login and Authentication Phases
% Receive login request message from Ui via open channel
2. State = 2 /\ Rcv(xor(H(IDi.Di), F(A'.F(Dgwn.P))).
          xor(IDj,F(A'.F(Dgwn.P))). F(A'.P).Ti'.
          F(A'.F(Dgwn.P)).G(A'.H(Ti'.F(A'.F(Dgwn.P)).
          H(H(IDi.Di).Dgwn)).F(A'.F(Dgwn.P)).Di)) =|>
State' := 4  /\ C' := new() /\ Tgwn' := new() /\ Cgwn':= F(C'.P)
 /\ Vgwn' := xor(H(H(H(IDi.Di).Dgwn).Ti'),H(F(A'.P).H(IDj.Dj).Tgwn'.Ti'))
          /\ Sgwn' := G(C'.H(H(H(IDi.Di).Dgwn).Ti').Cgwn'.Dgwn)
% Send authentication request message to SDj via open channel
          /\ Snd(Vgwn'.Tgwn'.Ti'.F(A'.P).Cgwn'.Sgwn')
% GWN has freshly generated the values c and Tgwn for SDj
         /\ witness (GWN, SDj, gwn_sdj_c, C')
         /\ witness (GWN, SDj, gwn_sdj_tgwn, Tgwn')
% GWN's acceptance of the values a and Ti generated for GWN by Ui
         /\ request(Ui, GWN, ui_gwn_a, A')
         /\ request(Ui, GWN, ui_gwn_ti, Ti')
end role
```

FIGURE 7. Role specification in HLPSL for the *GWN*.

```
role sensingdevice (Ui, GWN, SDj : agent, Snd, Rcv: channel(dy))
played_by SDj
def=
local State : nat, H, F, G: hash_func,
    P, Di, IDi, Ri, Dgwn, A, IDj, Ti, C, Tgwn : text,
    B, Tj, Dj, Bj, Sj, Kij, SKij : text
const sec3, gwn_sdj_c, gwn_sdj_tgwn, sdj_ui_b, sdj_ui_tj: protocol_id
init State := 0
transition
% Authentication Phase
% Receive authentication request message from GWN via open channel
1. State = 0 /\ Rcv(xor(H(H(H(IDi.Di).Dgwn).Ti'),H(F(A'.P).H(IDj.Dj).Tgwn'.Ti')).
          Tgwn'.Ti'.F(A'.P).F(C'.P).G(C'.H(H(H(IDi.Di).Dgwn).Ti').
          F(C'.P).Dgwn)) =|>
State' := 3  /\ B' := new() /\ Tj' := new() /\ Kij' := F(B'.F(A'.P))
         /\ SKij' := H(IDj.H(H(H(IDi.Di).Dgwn).Ti').Kij'.Ti'.Tj')
         /\ Bj' := F(B'.P) /\ secret(Dj, sec3, {SDj})
         /\ Sj' := G(B'.H(SKij'.Bj'.Dj))
% Send authentication reply to Ui via open channel
        /\ Snd(Bj'.Sj'.Tj')
% SDj has freshly generated the values b and Tj for Ui
     /\ witness (SDj, Ui, sdj_ui_b, B')
     /\ witness (SDj, Ui, sdj_ui_tj, Tj')
% SDj's acceptance of the values c and Tgwn generated for SDj by GWN
     /\ request(GWN, SDj, gwn_sdj_c, C')
     /\ request(GWN, SDj, gwn_sdj_tgwn, Tgwn')
end role
```

FIGURE 8. Role specification in HLPSL for the sensing device SD_j.

DETAILS: This explains the conditions where the scheme is safe or when attacks are possible or the reason for an inconclusive analysis.

BACK-END, GOAL and PROTOCOL: These indicate the backend used to analyze, the goal of the analysis and the name of the protocol respectively.

If an attack is found, the trace is printed in the standard Alice-bob format with a few statistics and comments.

```
role session (Ui, GWN, SDj : agent, SKugwn : symmetric_key)
def=
 local SN1, RV1, SN2, RV2, SN3, RV3 : channel (dy)
 composition
         user(Ui, GWN, SDj, SKugwn, SN1, RV1)
      /\ gwn(Ui, GWN, SDj, SKugwn, SN2, RV2)
      /\ sensingdevice(Ui, GWN, SDj, SN3, RV3)
end role

role environment()
def=
 const ui, gwn, sdj  : agent, skugwn : symmetric_key,
      h : hash_func, f, g : hash_func,
      p, didi, didj, ti, tgwn, tj: text,
      sec1, sec2, sec3, ui_gwn_a, ui_gwn_ti,
      gwn_sdj_c, gwn_sdj_tgwn, sdj_ui_b,
      sdj_ui_tj : protocol_id
intruder_knowledge = {h, f, g, p, didi, didj, ti, tgwn, tj}
composition
session(ui, gwn, sdj, skugwn)
   /\ session(i, gwn, sdj, skugwn)
   /\ session(ui, i, sdj, skugwn)
   /\ session(ui, gwn, i, skugwn)
end role

goal
% Confidentiality
secrecy_of sec1, sec2, sec3
% Authentication
authentication_on ui_gwn_a, ui_gwn_ti
authentication_on gwn_sdj_c, gwn_sdj_tgwn
authentication_on sdj_ui_b, sdj_ui_tj
end goal
environment()
```

FIGURE 9. Role specification in HLPSL for the session, goal and environment.

```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 C:\progra~1\SPAN\testsuite
  \results\auth.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 0.15s
 visitedNodes: 49 nodes
 depth: 6 plies
```

```
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL
PROTOCOL
 C:\progra~1\SPAN\testsuite
  \results\auth.if
GOAL
 As Specified
BACKEND
 CL-AtSe

STATISTICS

 Analysed   : 3 states
 Reachable  : 0 states
 Translation: 0.03 seconds
 Computation: 0.01 seconds
```

FIGURE 10. Analysis of simulation results using OFMC and CL-AtSe backends.

VII. PERFORMANCE COMPARISON

This section presents a performance comparison of the proposed scheme with other related authentication schemes [36]–[38] previously proposed for IoT applications. In Porambage *et al.*'s scheme [37], there are two protocols: protocol 1 allows only the legitimate members of the multicast group as eligible to continue the rest of the process of key derivation, and protocol 2 allows to establish a shared secret key among the multicast group.

The approximate time required for every operation and the terms used in calculating computational overhead are provided in Table 2. We use Table 2 for computational cost

TABLE 2. Approximate time required for various operations [35].

Notation	Description (time to compute)	Approx. computation time (seconds)
T_h	hash function	0.00032
T_{ecm}	ECC point multiplication	0.0171
T_{eca}	ECC point addition	0.0044

TABLE 3. Comparison of computation overheads of our scheme with related IoT schemes.

Protocol	User side	GWN/Base station side	Sensing device/ Sensor side	Total overhead
Our	$5T_{ecm}+5T_h$ $\approx 0.0871s$	$5T_{ecm}+4T_h$ $\approx 0.08678s$	$4T_{ecm}+3T_h$ $\approx 0.06936s$	$14T_{ecm}+12T_h$ $\approx 0.24324s$
[36]	$3T_h+2T_{ecm}$ $+T_{eca}$ $\approx 0.0396s$	–	$3T_h+2T_{ecm}$ $+T_{eca}$ $\approx 0.0396s$	$6T_h+4T_{ecm}$ $+2T_{eca}$ $\approx 0.0792s$
– Protocol-1 [37]	$4T_{ecm}+8T_h$ $+T_{eca}$ $\approx 0.0754s$	–	$11T_{ecm}+10T_h$ $+3T_{eca}$ $\approx 0.2045s$	$15T_{ecm}+18T_h$ $+4T_{eca}$ $\approx 0.2799s$
– Protocol-2 [37]	$3T_{ecm}+7T_h$ $+T_{eca}$ $\approx 0.0579s$	– –	$5T_{ecm}+7T_h$ $+2T_{eca}$ $\approx 0.0965s$	$8T_{ecm}+14T_h$ $+3T_{eca}$ $\approx 0.1544s$
[38]	$7T_h$ $\approx 0.00224s$	$5T_h$ $\approx 0.0016s$	$7T_h$ $\approx 0.00224s$	$19T_h$ $\approx 0.00608s$

computation required for the login and authentication phases. Table 3 shows the comparison of computational costs among the proposed scheme and other schemes [36]–[38]. From this table, it is observed that the computational cost of the proposed scheme is comparable to that for other schemes. The proposed scheme performs better than Porambage *et al.*'s scheme [37]. Though the proposed scheme requires more computational cost as compared to the schemes [36], [38], the proposed scheme offers more functionality features and better security as compared to the other schemes as shown in Table 5.

For comparing communication overheads among the proposed scheme and other related schemes, the following have been assumed:

- Sequence number, random nonce or time stamp is of length 32 bits.
- Hash function used is secure hash standard (SHA-1) [39]. Hence, hash digest length is 160 bits.
- Identity ID is of length 160 bits.
- As the security of 160-bit ECC cryptosystem is equivalent to that for 1024-bit RSA cryptosystem [40], an elliptic curve point $P = ((P)_x, (P)_y)$ requires $(160 + 160) = 320$ bits.

Table 4 show the communication overheads for all protocols during the login and authentication phases. The communication cost required by the proposed scheme is less than that for the schemes [37], [38]. However, our scheme needs more communication overhead as compared to that for Porambage *et al.*'s scheme [36]. It is justified as the proposed scheme offers more functionality features and better security as compared to the other schemes as shown in Table 5.

Finally, in Table 5, the availability of the desired functionality features in the existing schemes has been compared with the proposed scheme. The proposed scheme provides

TABLE 4. Comparison of communication overhead of our scheme with related IoT schemes.

Protocol	No. of messages	No. of bits
Our	3	2528
Porambage *et al.* [36]	4	1344
Porambage *et al.* [37]		
-Protocol-1	4	3360
-Protocol-2	2	1136
Turkanovic *et al.* [38]	4	2720

TABLE 5. Comparison of functionality features of the proposed scheme with related schemes.

Feature	Porambage *et al.* [36]	Porambage *et al.* [37]	Turkanovic *et al.* [38]	Our
FN_1	×	×	✓	✓
FN_2	×	✓	×	✓
FN_3	−	−	×	✓
FN_4	−	−	×	✓
FN_5	×	✓	✓	✓
FN_6	✓	×	✓	✓
FN_7	×	✓	×	✓
FN_8	×	✓	✓	✓
FN_9	×	×	✓	✓
FN_{10}	✓	✓	✓	✓
FN_{11}	✓	✓	✓	✓
FN_{12}	✓	✓	✓	✓
FN_{13}	−	−	×	✓
FN_{14}	✓	×	×	✓
FN_{15}	×	✓	✓	✓
FN_{16}	−	−	✓	✓
FN_{17}	−	×	×	✓
FN_{18}	×	×	×	✓
FN_{19}	×	×	×	✓

Note: FN_1: user anonymity property; FN_2: insider attack; FN_3: off-line password guessing attack; FN_4: stolen smart card attack; FN_5: denial-of-service attack; FN_6: known session key attack; FN_7: user impersonation attack; FN_8: man-in-the middle attack; FN_9: replay attack; FN_{10}: mutual authentication; FN_{11}: session key agreement; FN_{12}: forward secrecy; FN_{13}: stolen/lost device revocation; FN_{14}: untraceability property; FN_{15}: resilience against sensor node/sensing device capture attack; FN_{16}: GWN independent password update phase; FN_{17}: support biometric update phase; FN_{18}: provide security analysis using BAN logic; FN_{19} : provide formal security verification using AVISPA tool.
−: not applicable in a scheme; ×: insecure against a particular attack or does not support a particular feature; ✓: secure against a particular attack or supports a particular feature.

all the desired functionality features, while other schemes lack in key areas like providing user anonymity and security against impersonation and offline password guessing attacks. Also, a rigorous security analysis and formal security verification using the widely-accepted BAN logic and AVISPA tool, respectively, are not provided in other schemes.

VIII. PRACTICAL PERSPECTIVE: NS2 SIMULATION STUDY

In this section, we simulate our scheme using the widely-accepted network simulation tool, NS2 2.35 simulator [41] [42] on Ubuntu 14.04 LTS platform to measure the network performance parameters, such as throughput (in bps) and end-to-end delay (in seconds) to show the impact of the scheme.

A. SIMULATION PARAMETERS

The details of the parameters used in NS2 simulation are provided in Table 6. The network simulation time is taken as 1800 seconds (30 minutes). Both static and dynamic (mobile) types of users are considered in simulations. The speeds of the mobile users are considered as 2, 10 and 15 *mps*, respectively. Apart from these, all other standard parameters are taken for NS2 simulations.

TABLE 6. Various simulation parameters.

Parameter	Description
Platform	Ubuntu 14.04 LTS
Network scenarios	1, 2 and 3
Number of users (U_i)	$3, 5, 8$ for scenarios $1, 2, 3$
Number of gateway nodes (GWN)	1 for all scenarios
Number of smart devices (SD_j)	50 for all scenarios
Mobility	2 mps, 10 mps, 15 mps
Simulation time	1800 seconds

B. SIMULATION ENVIRONMENT

Three different network scenarios are used in the simulation. For all the scenarios, we have taken one *GWN* and 50 SD_js.

- *Scenario 1.* This scenario has three users (U_is): one is static and other two are moving with the speeds of 2 *mps* and 15 *mps*, respectively.
- *Scenario 2.* This scenario has five users (U_is): two are static and other three are moving with the speeds of 2 *mps*, 15 *mps* and 15 *mps*, respectively.
- *Scenario 3.* This scenario has eight users (U_is): four are static and other four are moving with the speeds of 2 *mps*, 2 *mps*, 10 *mps* and 15 *mps*, respectively.

Moreover, we assume that the hash output (if we use SHA-1 hash algorithm) and the identity have bit lengths 160 bits and 160 bits, respectively. In each scenario, messages communicated between different network entities are as follows: $\{DID_i', DID_j', A_i, T_i, r_i, s_i\}$ from U_i to GWN, $\{V_{GWN}, T_{GWN}, T_i, A_i, C_{GWN}, s_{GWN}\}$ from GWN to SD_j, $\{B_{SD_j}, s_{SD_j}, T_j\}$ from SD_j to U_i, which are of sizes 992 bits, 1024 bits and 512 bits, respectively.

C. SIMULATION RESULTS AND DISCUSSIONS

We have evaluated network performance parameters such as throughput (in bps) and end-to-end delay (in seconds) to measure the impact of the scheme.

1) IMPACT ON END-TO-END DELAY

End-to-end delay (*EED*) is computed as the average time taken by the data packets (messages) to arrive at the destination from the source. *EED* can be formulated as $\sum_{i=1}^{n_{pkt}}(T_{rec_i} - T_{send_i})/n_{pkt}$, where T_{rec_i} and T_{send_i} are the receiving and sending time of a packet i, respectively, and n_{pkt} the total number of packets. The *EED*s of the proposed scheme for different scenarios are given in Fig. 11. The *EED*s are 0.28683, 0.34588 and 0.36937 seconds for the network scenarios 1, 2 and 3, respectively. Further, note that the value of *EED* increases with the increasing number of users. This is because the increment in the number of users causes more messages to be exchanged, which further incurs congestion, and thus, *EED* increases in scenarios 2 and 3.

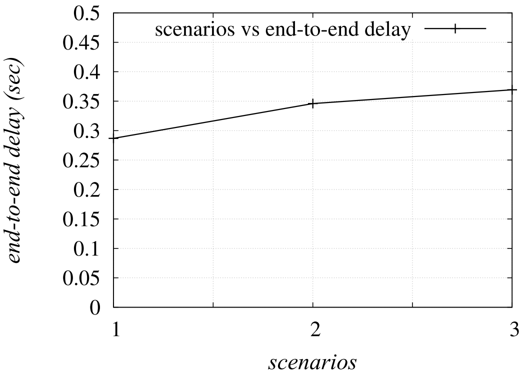

FIGURE 11. End-to-end delay of our scheme.

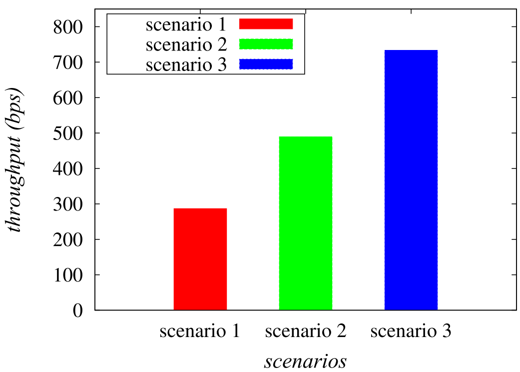

FIGURE 12. Throughput of our scheme.

2) IMPACT ON THROUGHPUT

Throughput is measured as the number of bits transmitted per unit time. Fig. 12 depicts the network throughput (in bps) of our scheme under different network scenarios. The throughput can be calculated as $\frac{n_r \times |pkt|}{T_d}$, where T_d is the total time (in seconds), $|pkt|$ the size of a packet, and n_r the total number of received packets. Note that we have considered the simulation time as $1800s$, which is the total time. Throughput values are 286.84, 489.51 and 733.49 *bps* for the scenarios 1, 2 and 3, receptively. The throughput increases in case of increment in number of users as the number of messages exchanged also increases.

IX. CONCLUSION

We have first discussed an authentication model for future IoT applications, and then the security challenges and requirements. We have presented a new signature-based user authenticated key agreement scheme to address the security challenges and requirements in IoT. The mutual authentication between a user and an accessed sensing device is proved using the broadly-accepted BAN logic. We have also shown the security of the proposed scheme informally and the formal security verification using the widely-accepted AVISPA tool. A rigorous security analysis reveals that the proposed scheme can be protected against various known attacks by an adversary. Various network parameters are measured through a rigorous simulation using the widely-used NS2 simulator. The proposed scheme is also efficient in computation and communication, and these are comparable with other existing approaches. High security, efficient computational and communication costs along with additional functionality features show that the proposed scheme is suitable for practical applications in IoT environment as compared to other related schemes.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers and the Associate Editor for their valuable feedback on the paper which helped us to improve its quality and presentation.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generat. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[3] 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.

[4] 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.

[5] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology*, vol. 1666. Berlin, Germany: Springer-Verlag, 1999, pp. 388–397.

[6] N. Koblitz, "Elliptic curves cryptosystems," *Math. Comput.*, vol. 48, pp. 203–209, Sep. 1987.

[7] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle, "DTLS based security and two-way authentication for the Internet of Things," *Ad Hoc Netw.*, vol. 11, no. 8, pp. 2710–2723, Sep. 2013.

[8] D. Liu, P. Ning, S. Zhu, and S. Jajodia, "Practical broadcast authentication in sensor networks," in *Proc. 2nd Annu. Int. Conf. Mobile Ubiquitous Syst. Netw. Serv. (MobiQuitous)*, San Diego, CA, USA, 2005, pp. 118–129.

[9] X. Cao, W. Kou, L. Dang, and B. Zhao, "IMBAS: Identity-based multi-user broadcast authentication in wireless sensor networks," *Comput. Commun.*, vol. 31, no. 4, pp. 659–667, Mar. 2008.

[10] D. J. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography," in *Proc. 1st IEEE Int. Conf. Sensor Ad Hoc Commun. Netw.*, Oct. 2004, pp. 71–80.

[11] K. Ren, S. Yu, W. Lou, and Y. Zhang, "Multi-user broadcast authentication in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 8, pp. 4554–4564, Oct. 2009.

[12] S. Yamakawa, Y. Cui, K. Kobara, and H. Imai, "Lightweight broadcast authentication protocols reconsidered," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Budapest, Hungary, Apr. 2009, pp. 1–6.

[13] P. Szczechowiak, L. B. Oliveira, M. Scott, M. Collier, and R. Dahab, "NanoECC: Testing the limits of elliptic curve cryptography in sensor networks," in *Proc. 5th Eur. Conf. Wireless Sensor Netw. (EWSN)*, Bologna, Italy, 2008, pp. 305–320.

[14] D. He, S. Zeadally, N. Kumar, and J. H. Lee, "Anonymous authentication for wireless body area networks with provable security," *IEEE Syst. J.*, to be published, doi: 10.1109/JSYST.2016.2544805.

[15] D. He and S. Zeadally, "An analysis of RFID authentication schemes for Internet of Things in Healthcare environment using elliptic curve cryptography," *IEEE Internet Things J.*, vol. 2, no. 1, pp. 72–83, Feb. 2015.

[16] 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.

[17] A. K. Das, S. Kumari, V. Odelu, X. Li, F. Wu, and X. Huang, "Provably secure user authentication and key agreement scheme for wireless sensor networks," *Secur. Commun. Netw.*, vol. 9, no. 16, pp. 3670–3687, 2016.

[18] S. H. Seo, J. Won, S. Sultana, and E. Bertino, "Effective key management in dynamic wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 371–383, Feb. 2015.

[19] D. He and S. Zeadally, "Authentication protocol for an ambient assisted living system," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 71–77, Jan. 2015.

[20] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: Security protocols for sensor networks," *Wireless Netw.*, vol. 8, no. 5, pp. 521–534, 2002.
[21] K. Ren, W. Lou, K. Zeng, and P. J. Moran, "On broadcast authentication in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 11, pp. 4136–4144, Nov. 2007.
[22] D. Liu and P. Ning, *Multi-Level MicroTESLA: A Broadcast Authentication System for Distributed Sensor Networks*. Raleigh, NC, USA: IEEE, 2003.
[23] J. Shaheen, D. Ostry, V. Sivaraman, and S. Jha, "Confidential and secure broadcast in wireless sensor networks," in *Proc. 18th IEEE Int. Symp. Pers., Indoor Mobile Radio Commun.*, Athens, Greece, Sep. 2007, pp. 1–5.
[24] T. Wu *et al.*, "A fast and efficient source authentication solution for broadcasting in wireless sensor networks," in *New Technologies, Mobility and Security*. Nashville, TN, USA: Institute for Software Integrated Systems, 2007, pp. 53–63.
[25] A. Perrig, "The BiBa one-time signature and broadcast authentication protocol," in *Proc. 8th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, 2001, pp. 28–37.
[26] S.-M. Chang, S. Shieh, W. W. Lin, and C.-M. Hsieh, "An efficient broadcast authentication scheme in wireless sensor networks," in *Proc. ACM Symp. Inf., Comput. Commun. Secur.*, New York, NY, USA, Sep. 2006, pp. 311–320.
[27] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Advances in Cryptology—CRYPTO*. Berlin, Germany: Springer, 1988, pp. 369–378.
[28] L. Reyzin and N. Reyzin, "Better than BiBa: Short one-time signatures with fast signing and verifying," in *Proc. 7th Austral. Conf., Inf. Secur. Privacy, (ACISP)*, Melbourne, Australia, 2002, pp. 144–153.
[29] P. Rohatgi, "A compact and fast hybrid signature scheme for multicast packet authentication," in *Proc. 6th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, 1999, pp. 93–100.
[30] A. K. Das, P. Sharma, S. Chatterjee, and J. K. Sing, "A dynamic password-based user authentication scheme for hierarchical wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 35, no. 5, pp. 1646–1656, 2012.
[31] S. Kumari *et al.*, "On the design of a secure user authentication and key agreement scheme for wireless sensor networks," *Concurrency Comput. Pract. Exper.*, to be published, doi: 10.1002/cpe.3930.
[32] AVISPA. *Automated Validation of Internet Security Protocols and Applications*, accessed on Jan. 2017. [Online]. Available: http://www.avispa-project.org/
[33] D. von Oheimb, "The high-level protocol specification language HLPSL developed in the eu project AVISPA," in *Proc. 3rd APPSEM II Workshop Appl. Semantics (APPSEM)*, Frauenchiemsee, Germany, 2005, pp. 1–17.
[34] AVISPA. *SPAN, the Security Protocol Animator for AVISPA*, accessed on Jan. 2017. [Online]. Available: http://www.avispa-project.org/
[35] V. Odelu, A. K. Das, and A. Goswami, "An efficient biometric-based privacy-preserving three-party authentication with key agreement protocol using smart cards," *Secur. Commun. Netw.*, vol. 8, no. 18, pp. 4136–4156, 2015.
[36] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase authentication protocol for wireless sensor networks in distributed IoT applications," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Istanbul, Turkey, Sep. 2014, pp. 2728–2733.
[37] P. Porambage, A. Braeken, C. Schmitt, A. Gurtov, M. Ylianttila, and B. Stiller, "Group key establishment for enabling secure multicast communication in wireless sensor networks deployed for IoT applications," *IEEE Access*, vol. 3, pp. 1503–1511, Apr. 2015.
[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] "Secure hash standard," U.S. Dept. Commerce, Nat. Inst. Standards Technol. (NIST), Gaithersburg, MD, USA, Tech. Rep. FIPS PUB 180-1, Apr. 1995.
[40] S. Vanstone, "Responses to NIST's proposal," *Commun. ACM*, vol. 35, no. 7, pp. 50–52, 1992.
[41] *The Network Simulator-NS-2*, accessed on Apr. 2016. [Online]. Available: http://www.isi.edu/nsnam/ns/
[42] J. Wang. *NS-2 Tutorial*, accessed on Apr. 2016. http://www.cs.virginia.edu/~757/slidespdf/cs757-ns2-tutorial1.pdf

SRAVANI CHALLA received the M.Sc. (Tech) degree in information systems from the Birla Institute of Technology and Science (BITS), Pilani, India. She is currently pursuing the M.S. degree in computer science and engineering with the International Institute of Information Technology, Hyderabad, India. She has authored three journal papers in the above areas. Her research interests include cryptography and network security.

MOHAMMAD WAZID received the M.Tech. degree in computer network engineering from Graphic Era University, Dehradun, India. He is currently pursuing the Ph.D. degree with the International Institute of Information Technology, Hyderabad, India. He has authored over 40 papers in international journals and conferences in the above areas. His current research interests include cryptography and security in wireless sensor network, vehicular ad-hoc network, and cloud computing. He was a recipient of the University Gold Medal and the Young Scientist Award from UCOST, Department of Science and Technology, Government of Uttarakhand, India.

ASHOK KUMAR DAS 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 Assistant Professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. He has authored over 125 papers in international journals and conferences in the above areas. His current research interests include cryptography, wireless sensor network security, hierarchical access control, data mining, security in vehicular ad hoc networks, smart grid and cloud computing, and remote user authentication. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He is on the editorial board of the *KSII Transactions on Internet and Information Systems*, and the *International Journal of Internet Technology and Secured Transactions* (Inderscience), and a Guest Editor of the *Computers and Electrical Engineering* (Elsevier) for the special issue on big data and IoT in e-healthcare, and has served as a Program Committee Member in many international conferences.

NEERAJ KUMAR (M'16) received the Ph.D. degree in computer science and engineering from Shri Mata Vaishno Devi University, Katra, India, in 2009. He was a Post-Doctoral Research Fellow with Coventry University, Coventry, U.K. He is currently an Associate Professor with the Department of Computer Science and Engineering, Thapar University, Patiala, India. He has authored over 160 technical research papers published in leading journals and conferences from the IEEE, Elsevier, Springer, and Wiley. Some of his research findings are published in top cited journals such as the IEEE Transactions on Industrial Electronics, the IEEE Transactions on Dependable and Secure Computing, the IEEE Transactions on Intelligent Transportation Systems, the IEEE Transactions on Consumer Electronics, the IEEE Network, IEEE Communications, IEEE Wireless Communications, the IEEE Internet of Things Journal, the IEEE Systems Journal, the *Future Generation Computer Systems*, the *Journal of Network and Computer Applications*, and the *Computer Communications*. He has guided many research scholars leading to Ph.D. and M.E./M.Tech degrees.

ALAVALAPATI GOUTHAM REDDY received the master's degree in computer science and engineering from Christ University, India, in 2013, and the Ph.D. degree from the School of Computer Science and Engineering, Kyungpook National University, South Korea. His primary research interests revolve around cryptography, authentication technologies, and information security. He is a student member of the ACM.

EUN-JUN YOON received the Ph.D. degree in computer engineering from Kyungpook National University, South Korea, in 2006. He is currently a Professor with the Department of Cyber Security, Kyungil University, South Korea. He has authored 75 conference proceedings and 50 journal publications. His research interests are cryptography, authentication technologies, smart card security, multimedia security, network security, mobile communications security, and steganography.

KEE-YOUNG YOO (M'16) received the M.S. degree in computer engineering from KAIST, South Korea, in 1978, and the Ph.D. degree in computer science from the Rensselaer Polytechnic Institute, USA, in 1992. He is currently a Professor with the School of Computer Science and Engineering, Kyungpook National University, South Korea. His areas of expertise include cryptography, steganography, wireless mesh networks, and RFID security. He has authored over 200 conference proceedings and 195 journal publications.