# An Interoperable ECC based Authentication and Key Agreement Scheme for IoT Environment

Asit Sahoo
*Computer Science And Engineering*
*NIT Rourkela, India*
618cs6004@nitrkl.ac.in

Shreeya Swagatika Sahoo
*Computer Science And Engineering*
*NIT Rourkela, India*
shreeya.swagatika@gmail.com

Sampa Sahoo
*Computer Science And Engineering*
*NIT Rourkela, India*
sampaa2004@gmail.com

Bibhudatta Sahoo
*Computer Science And Engineering*
*NIT Rourkela, India*
bibhudatta.sahoo@gmail.com

Ashok Kumar Turuk
*Computer Science And Engineering*
*NIT Rourkela, India*
akturuk@gmail.com

***Abstract*—With the advancement in technology, the Internet of Things (IoT) systems has been blooming in leap and bounds. It aims at connecting things such as vehicles, hospitals, industries, and consumers through the Internet. The increase in a number of IoT devices and the heterogeneity of their network connection is increasing day by day; it has given rise to several challenges such as authenticity, cost and usability, scalability, interoperability, mobility, and many more. Interoperability is one of such security challenges which has the ability for systems or components to communicate with each other, regardless of their manufacturer or technical specifications. Further, security and privacy issue is also a significant concern in the IoT environment. So, the IoT system must be secured, and proper authentication schemes have to be integrated to restrain the unauthorized access. Besides, the limited computation capability of the sensor generates the need for the light-weight authentication protocol. In this regard, this paper discusses an interoperable light-weight authentication protocol for the IoT system. To demonstrate the feasibility of the scheme, we employed a widely used formal verification tool Proverif for correctness proof of the scheme. Additionally, informal security analysis demonstrates that the scheme is secure against most of the known attacks.**

***Index Terms*—Authentication, ECC, IoT, Proverif.**

## I. Introduction

IoT is defined as an interconnected network architecture of self-configured and intelligent nodes (or smart devices) that can interact through the Internet. It helps to enhance the efficiency of the system at a lower cost. According to a report presented in [1] "By 2025 Internet nodes may reside in every day things—food packages, furniture, paper documents, and more. The IoT devices are connected to the Internet through heterogeneous access and network technology. This claim can be justified by the report presented in [2] which states that, (i) number of connected devices will rise to 50 billion by 2020 (predicted by CISCO), (ii) global spending on IoT will rise to the US $1.7 trillion by 2020 (prediction by IDC) (iii) 90% of cars will be connected to Internet by 2020 (prediction by Telefonica) (iv) a quarter billion vehicles will be connected to the Internet (estimation by Gartner). Increasing in IoT devices have raised various issues for the system such as flexibility, interoperability, minimizing the cost of the system, maximizing utilization of the resources, mobility, etc. Further, the integration of IoT devices and cloud servers depends highly on "How security issues such as authentication and data privacy are handled."

The Kaspersky report mentioned in [3] says that most of the attacks were implanted using routers, which act as a gateway for the IoT devices. The addition of resourceful gateway will allow quick on-demand delivery of data or information and take care of most of the processing. The author in [4] pointed out that attackers mainly targets the everyday consumer gadgets such as home-networking routers, connected multimedia centers, televisions, refrigerators, etc. Further, most of the attacks encountered in the IoT system are possible because of the lack of proper authentication between communicating entities (e.g., sensors, gateway, servers, etc.).

Authentication among devices and users backs to transmit data to the legitimate entity without the concern of being manipulated by unauthorized personnel. A secure authentication protocol can restrict access of the IoT system to the authorized entities resulting in enhancement of the security and trust of various users availing the services for the system. As the sensors in the IoT system are resource-constrained in terms of power computing, memory requirement, etc., a lightweight security solution is needed. The small key size and computation efficiency of Elliptic Curve Cryptography (ECC) make it preferable over other Public-Key Cryptography (PKC) for better security solutions.

Large use of heterogeneous devices makes communication unfavorable among themselves. Interoperability is responsible for seamless communication among heterogeneous devices. The IEEE defines interoperability as "the ability of two or more systems or components to exchange information and to use the information that has been exchanged" [6]. Authors in [7] have highlighted the issues due to the lack of interoperability in the IoT platform and proposed a complete smartphone-centric gateway application. There may be gateways and sensors which support interoperable operations to communicate, but a little work has been done on securing communication

among the entities, keeping the interoperability intact. In this paper, a secure and interoperable lightweight authentication protocol is design for an IoT system. The proposed protocol is lightweight as it uses operations such as hash functions and XOR with low computation overhead. Further, the proposed scheme uses ECC for better security solutions. The formal verification and correctness of the proposed scheme are performed using the Proverif tool, which confirms the session key security of the scheme. In addition, the informal security analysis of the proposed scheme guarantees that the scheme is robust against several known attacks.

The paper is organized as follows. The next Section demonstrates a brief review of related works. The proposed IoT network model and scheme are discussed in Section III and section IV respectively. Section V demonstrates the security analysis and simulation of the proposed scheme using Proverif. The paper is finally concluded in section VI.

## II. Related Work

To improve the security in the IoT system, various authentication and key agreement schemes have been suggested. A review of the related scheme is presented in the following. Chuang *et al.* [8] proposed authentication and key agreement scheme between multi-server and user. Amin *et al.* [9] proved the scheme to be vulnerable to user impersonation attack, session key disclosure attack, and proposed an authentication scheme that addresses these attacks. Kalra *et al.* [10] proposed an ECC based mutual authentication protocol for IoT devices and cloud servers using encrypted cookies. Kumari *et al.* [11] proved that the scheme [10] is vulnerable to offline password guessing attack, insider attack and lacks device anonymity, session key agreement, and mutual authentication. Further, they designed a scheme that can resist the known attacks. Liu *et al.* [12] proposed a user authentication scheme using bilinear pairing to establish a secure connection between the user and sensor nodes. Challa *et al.* [13] proved the scheme prone to stolen smart card attack, offline password guessing attack, user impersonation attack. It further proved that the scheme fails to provide user anonymity and mutual authentication. In 2018, Challa *et al.* proposed a user authentication scheme using ECC, claiming it to be resistant from the known network attacks.

Farash *et al.* [14] proposed a user authentication and key agreement scheme for heterogeneous wireless sensor network (WSN) for IoT. Amin *et al.* [15] proved that the scheme could not withstand the user anonymity attack, user impersonation attack, and known session-specific temporary information attack. They even pointed out that the architecture followed in the scheme was not energy efficient. Amin *et al.* proposed a 3-factor user authentication scheme in WSN on his proposed architecture. Further, Sharif *et al.* [16] proved that the protocol was vulnerable to replay attacks and could not provide perfect forward secrecy. They proposed a secure key agreement protocol between the user, gateway, and the sensor claiming it to be resistant from the above-mentioned attacks. In this paper [17], a lightweight biometric-based authentication and key agreement scheme are proposed for an IoT system. In [18], authors have designed a protocol which employs gateway node-based architecture for the IoT environment, which requires the user first to register itself through the gateway node. Chuang *et al.* [19] proposed an authentication scheme between sensor node and the gateway. It calculates the battery capacity of the sensor and uses it as one of the parameters for the authentication scheme. It claims to be resistant from replay attacks, impersonation attacks, the man in the middle attack. It further claims to provide mutual authentication and perfect forward secrecy. Zhou *et al.* [20] proposed a lightweight authentication scheme for the user and cloud server using a control server. The scheme claims to be resistant against user offline guessing attack, insider attack, desynchronization attack. Also, it provides mutual authentication and user anonymity. Sharma *et al.* [21] proposed a lightweight multi-factor remote user authentication scheme between the user and the cloud server. It claims to be resistant from major network attacks such as impersonation attack, the man in the middle attack, offline password guessing attack, replay attack, etc.

Throughout our survey, we find most of the authentication scheme requires the dependency of a specific gateway for authentication of a fixed number of sensor nodes as it stores prior information shared between them in its memory before its deployment in the network. However, it leads to a loss of interoperability among the devices. Secondly, many researchers focused on the authentication scheme only between the users and the cloud servers. A very few works have been done on developing the security protocol for the whole IoT system. The above motivates us to develop an interoperable security protocol to enhance the security of the entities in the IoT system. Our proposed scheme reduces the network cost (building, maintenance), providing secure communication among entities. In this scheme, users will have the sole access to register the sensors with the gateway, unlike the traditional model where a fixed number of sensors were pre-registered with the gateway, eliminating interoperability in the network.

## III. The Proposed Network Model

The layered architecture of the IoT network model used in this paper is shown in Fig. 1. The layers of the system are the perception layer, network layer, and application layer. The perception layer comprises of different sensors/things to collect real-time data from the environment. The heterogeneity of the sensors affects its security level. The network layer acts as a bridge between the perception layer and the application layer. It transfers the data generated in the perception layer via the gateway to the application layer for further processing. Earlier architecture for IoT system uses a specific gateway for a specific set of sensors. For instance, sensors S1 - S7 need a gateway which can handle data generated by it. Similarly, sensors S8 - S13 need a specific gateway for it, and so on. Moreover, sensor manufactured by different manufacturers has compatibility or interoperability issues. In this context, we have used a single gateway that can connect sensors irre-

spective of type and manufacturers to ensure an interoperable environment. The application layer includes cloud platforms, middleware, data analysis, expert system, etc. It is responsible for processing the data collected from the lower layer and giving services to the end-users.

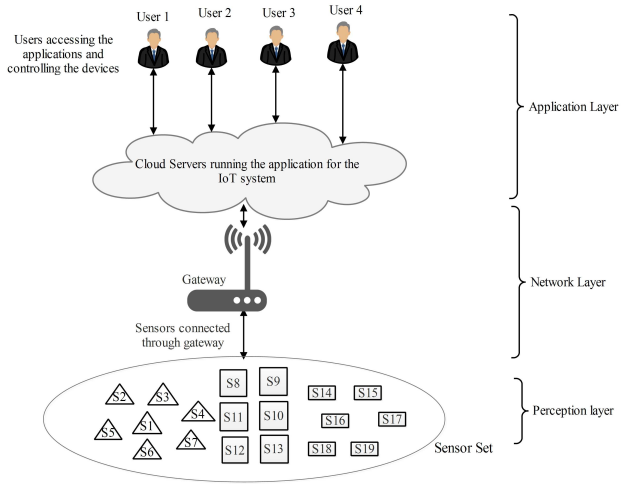


Fig. 1. Three Layered IoT Network Model

The overall working of the proposed protocol is depicted in Fig. 2. Our protocol consists of the following phases: system setup phase, user registration phase, user login phase, gateway registration phase, sensor registration phase, and authentication and key agreement phase. Table I mentions the notations used in the proposed protocol. A new user registers with the server using his/her identity, password, and biometric. The server provides a smart card to the user by embedding some parameters in it. Then, the user login into the system, using his / her credentials, and its existence is checked by the cloud server. If the user identity is not present, then the session is terminated. Otherwise, the user inputs gateway identity and the total number of sensors in the system known to him/her. Then, the input gateway identity and the registered gateway identity are compared by the cloud server for the gateway registration phase. Initially, the total number of registered sensors and the identity of the gateway are initialized with null values. With the execution of different phases, the values get updated and stored in the cloud server. If the input gateway identity and the registered gateway identity match, then the gateway is already registered; otherwise, it will go to the gateway registration phase.

In the gateway registration phase, the gateway sends its identity to the user, and the user validates the gateway. On successful validation, the gateway computes some data and sends those data to the server for its registration. The server validates the gateway and updates the registered gateway identity with the received gateway identity. Then, it starts the sensor registration phase; otherwise, it terminates the session. Further, if the gateway is already registered, then it will go to the sensor registration phase directly.

In the sensor registration phase, the user sends its own identity and the sensor identity to the registered gateway. It further checks the validity of the sensor and store it in its memory. Later, the gateway request the cloud server for further processing. The server updates the number of registered sensors by incrementing it by one and further checks if the number of sensor inputs by the user is greater than the number of the registered sensors. If the condition holds, it deploys the sensor registration phase. Else, it starts the authentication and key agreement phase.

In this phase, the server authenticates the gateway of the logged-in user. On successful authentication, gateway authenticates the server, and session keys are negotiated between them. Further, gateway and sensor mutually authenticate themselves, and the session key is exchanged between them. Thus, all the entities are authenticated in the system, and the session keys are negotiated between them. The employment of sensor and gateway addition phase in the proposed scheme allows the addition of sensors and gateway any time, making the scheme scalable.

TABLE I
NOTATIONS USED

| Notation | Description |
|---|---|
| $U_i$ | $i^{th}$ User |
| $SM$ | Cloud Server |
| $GW$ | Gateway |
| $S_l$ | $l^{th}$ Sensor |
| $ID_i$ | Identity of $i^{th}$ User |
| $PWD_i$ | Password of $i^{th}$ User |
| $BIO_i$ | Bio metric of $i^{th}$ User |
| $GID_k$ | Identity of Gateway |
| $SID_l$ | Identity of $l^{th}$ Sensor |
| $E_k/D_k$ | AES Encryption/Decryption with key k |
| $NEWC_i$ | Total number of sensor known to user including the unregistered ones. |
| $CURRC_i$ | Total number of registered sensors |
| $NEWG_i$ | Gateway id known to user |
| $CURRG_i$ | Current id of registered gateway. |
| $w_k$ | Gateway secret key. |
| $x$ | Shared secret key between gateway and Cloud server. |
| $TS_{curr}$ | Current timestamp |

## IV. THE PROPOSED SCHEME

This section puts forward a provably secure, ECC based interoperable authentication scheme for the proposed IoT architecture.

### A. System setup phase

1) It is assumed that the system uses an elliptic curve $E_P(a,b)$ over a finite field $Z_P$, $P$ being a large prime, a base point w of order n over $E_P(a,b)$ where $4a^3 + 27b^2 \neq 0(\mathrm{mod}P)$ and $n.w = o$, $o$ being the point of infinity.
2) Gateway is considered more computationally efficient than sensors and has a memory to store the list of the registered sensors.
3) Sensors and gateways store their respective identities in their memory before the deployment.
4) Multiple sensors (up to a certain limit) can be managed by a single gateway.
5) It is assumed that the user has the knowledge of $GID_k$ and all the $SID_k$ of his/her system.

### B. User registration phase

$U_i$ submits his/her credentials such as $ID_i$, $PWD_i$ and $BIO_i$ to $SM$. $SM$ register $U_i$ using the following steps

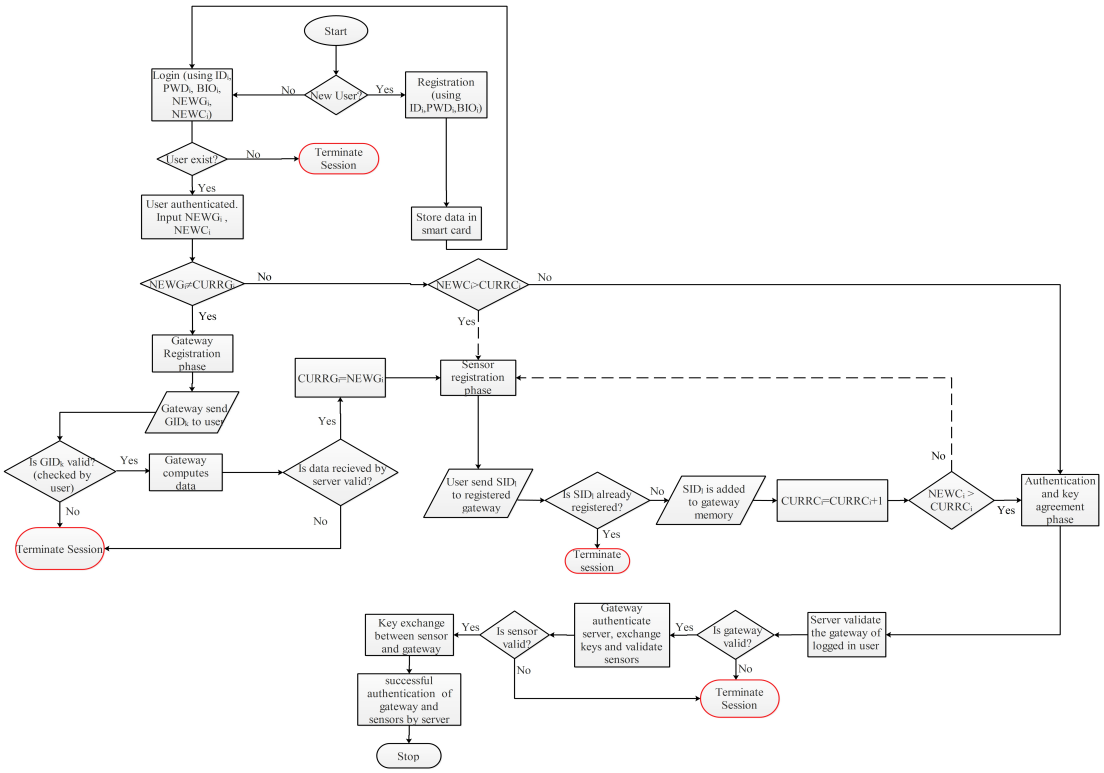


Fig. 2. Working of the phases in the protocol

through a secure channel. The workflow of the user registration phase is shown in Table II.

1) $U_i$ selects $ID_i$, $PWD_i$, $BIO_i$ and a random number $N_1$. It computes $R_{bio_i} = h(ID_i||PWD_i||BIO_i||N_1)$, $Y_i = N_1 \oplus h(ID_i||PWD_i)$.
2) He/She then selects another random number $N_2$ and generates $D_i$ as $D_i = N_2.w = (D_x, D_y)$. Then, $U_i$ computes $B_i = N_2 \oplus h(ID_i||PWD_i)$, $H_{b_i} = BIO_i \oplus h(ID_i||D_i)$ and send a request message $\{Y_i, D_i, ID_i, R_{bio_i}\}$ to the $SM$ over a secure channel.
3) $SM$ stores $\{Y_i, R_{bio_i}, ID_i\}$ in its database. It selects a random number $N_3$ and compute $E_i = N_3.w = (E_x, E_y)$, $RK_{im} = N_3.D_i$, and $TEM_i = ID_i \oplus D_i$. $SM$ stores $\{Rk_{im}, TEM_i\}$ in its database corresponding to the received $ID_i$. $SM$ embeds computed parameter $E_i$ into smart card ($SC_i$) and send it to $U_i$.
4) $U_i$ receives the smart card and store $\{B_i, N_1, H_{b_i}\}$ into the smart card. Finally $SC_i$ contains $\{B_i, E_i, N_1, H_{b_i}\}$.

### C. User login phase

In this phase, the user has to log in before accessing the resources of the system. The login phase is summarized in Table III.

1) $U_i$ inserts his/her smart card to the card reader and inputs his/her $\{ID_i, PWD_i, BIO_i, NEWC_i, NEWG_i\}$.
2) The card computes $N_2^* = B_i \oplus h(PWD_i||ID_i)$, $D_i' = N_2^*.w = (D_x', D_y')$, $H_{b_i}^* = BIO_i \oplus h(ID_i||D_i')$. Further, it checks $H_{b_i}^* \stackrel{?}{=} H_{b_i}$ holds true, $SC_i$ computes $RK_{im}^* = N_2^*.E_i$, $TEM_i = ID_i \oplus D_i'$. It uses the computed parameters to encrypt the inputs as $LINF_i = E_{RK_{im}^* \oplus D_i'}(ID_i, PWD_i, BIO_i)$.
3) $SC_i/U_i$ transmit $\{LINF_i, TEM_i, TS_1\}$ to the cloud server via a public channel.
4) $SM$ receives the login request and verifies the validity of timestamp $TS_1$ by checking $|TS_{curr} - TS_1| \leq \Delta T$. If the condition is not satisfied, the session is terminated and login request is rejected. Otherwise, $SM$ checks its database for $TEM_i$ and it retrieves the corresponding $\{RK_{im}, ID_i\}$.
5) $SM$ computes $D_i'' = ID_i \oplus TEM_i$ and decrypts the received $LINF_i$ using the key $RK_{im} \oplus D_i''$. It computes

TABLE II
USER REGISTRATION PHASE

| User($U_i$) | | Cloud Server($SM$) |
|---|---|---|
| Select a random number $N_1$ | | |
| $R_{bio_i} = h(ID_i\|\|PWD_i\|\|BIO_i\|\|N_1)$ | | |
| $Y_i = N_1 \oplus h(ID_i\|\|PWD_i)$ | | |
| Select a random number $N_2$ | | |
| $D_i = N_2.w = (D_x, D_y)$ | | |
| $B_i = N_2 \oplus h(PWD_i\|\|ID_i)$ | | |
| $H_{b_i} = BIO_i \oplus h(ID_i\|\|D_i)$ | | |
| | $\dashrightarrow \{Y_i, D_i, ID_i, R_{bio_i}\}$ | |
| | | Store $Y_i, R_{bio_i}, ID_i$ |
| | | Select a random number $N_3$ |
| | | $E_i = N_3.w = (E_x, E_y)$ |
| | | $RK_{im} = N_3.D_i$ |
| | | $TEM_i = ID_i \oplus D_i$ |
| | | Store $Rk_{im}, TEM_i$ |
| | $\dashleftarrow \{E_i\}$ | |
| Remember $B_i, E_i, N_1, H_{b_i}$ | | |
| | | Secure channel $- - - \rightarrow$ |

TABLE III
USER LOGIN PHASE

| User($U_i$) | | Cloud Server($SM$) |
|---|---|---|
| Input $ID_i, PWD_i, BIO_i, NEWC_i, NEWG_i$ | | |
| $N_2^* = B_i \oplus h(PWD_i\|\|ID_i)$ | | |
| $D_i' = N_2^*.w = (D_{i_x}', D_{i_y}')$ | | |
| $H_{b_i}^* = BIO_i \oplus h(ID_i\|\|D_i')$ | | |
| if($H_{b_i}^* == H_{b_i}$) | | |
| $RK_{im}^* = N_2^*.E_i$ | | |
| Choose the current timestamp $TS_1$ | | |
| $LINF_i = E_{RK_{im}^* \oplus D_i'}(PWD_i, BIO_i)$ | | |
| $TEM_i = ID_i \oplus D_i'$ | | |
| | $\xrightarrow{\{LINF_i, TEM_i, TS_1\}}$ | |
| | | $\|TS_{curr} - TS_1\| \leq \Delta T$ |
| | | Check database for $TEM_i'$ and its corresponding $RK_{im}, ID_i$ |
| | | $D_i'' = ID_i \oplus TEM_i$ |
| | | $D_{RK_{im} \oplus D_i''}(LINF_i) = (PWD_i, BIO_i)$ |
| | | $N_1^* = h(ID_i\|\|PWD_i) \oplus Y_i$ |
| | | $R_{bio_i}^* = h(ID_i\|\|PWD_i\|\|BIO_i\|\|N_1^*)$ |
| | | If ($R_{bio_i}^* == R_{bio_i}$), then |
| | | Select a random number $N_4$ |
| | | $TP_i = R_{bio_i}^* \oplus N_4$ |
| | | $SUK_{im} = h(ID_i\|\|N_1^*\|\|N_4)$ |
| | | Choose the current Timestamp $TS_2$ |
| | | $VSU_{im} = h(SUK_{im}\|\|TS_2)$ |
| | | Else, Abort session |
| | $\xleftarrow{\{TP_i, VSU_{im}, TS_2\}}$ | |
| $\|TS_{curr} - TS_2\| \leq \Delta T$ | | |
| $N_4^* = TP_i \oplus h(ID_i\|\|PWD_i\|\|BIO_i\|\|N_1)$ | | |
| $SUK_{im}^* = h(ID_i\|\|N_1\|\|N_4^*)$ | | |
| $VSU_{im}^* = h(SUK_{im}^*\|\|TS_2)$ | | |
| if($VSU_{im} == VSU_{im}^*$) then | | |
| Input $SID_l$ | | |
| Select a random value $N_5$ | | |
| $T_{km} = h(N_5\|\|TS_2\|\|ID_i)$ | | |
| Store $T_{km}$ | | |
| $F_i = E_{SUK_{im}^*}(NEWC_i, NEWG_i, T_{km}, SID_l)$ | | |
| Choose the current timestamp $TS_3$ | | |
| $C_i = h(N_1\|\|TS_3\|\|D_i')$ | | |
| | $\xrightarrow{\{F_i, C_i, TS_3\}}$ | |
| | | $\|TS_{curr} - TS_3\| \leq \Delta T$ |
| | | $C_i' = h(N_1^*\|\|TS_3\|\|D_i'')$ |
| | | if ($C_i == C_i'$) then |
| | | $D_{SUK_{im}}(F_i) = \{NEWC_i, NEWG_i, T_{km}, SID_l\}$ |
| | | Temporarily store $SID_l, T_{km}$ for the session |
| | | if($NEWG_i \neq CURRG_i$) |
| | | Gateway Registration Phase |
| | | else if($NEWC_i > CURRC_i$) |
| | | Sensor Registration Phase |
| | | Else, Authentication and key agreement Phase |
| | | Insecure channel$\longrightarrow$ |

$N_1^* = h(ID_i||PWD_i) \oplus Y_i$ and employs $N_1^*$ to compute $R_{bio_i}^* = h(ID_i||PWD_i||BIO_i||N_1^*)$. If the condition $R_{bio}^* \stackrel{?}{=} R_{bio}$ doesn't hold true, session is aborted. Else, it chooses a random number $N_4$ to compute $TP_i = R_{bio_i}^* \oplus N_4$, the session key $SUK_{im} = h(ID_i||N_1^*||N_4)$, and $VSU_{im} = h(SUK_{im}||TS_2)$. Then it send message $\{TP_i, VSU_{im}, TS_2\}$ to $U_i/SC_i$.

6) The message $\{TS_2, TP_i, VSU_{im}\}$ is received by $U_i$ and then it verifies the timestamp $|TS_{curr} - TS_2| \leq \Delta T$. If the verification is unsuccessful, the session is aborted by the $U_i$. Else, $U_i$ compute $N_4^* = TP_i \oplus h(ID_i||PWD_i||BIO_i||N_1)$, session key $SUK_{im}^* = h(ID_i||N_1||N_4^*)$, $VSU_{im}^* = h(SUK_{im}^*||TS_2)$ and checks the condition $VSU_{im}^* \stackrel{?}{=} VSU_{im}$. If the condition doesn't holds true then, the user terminates the session. Else $U_i$ inputs sensor identity $SID_l$, it wants to send or receive data. Further, it computes $T_{km} = h(N_5||TS_2||ID_i)$, store $T_{km}$ and encrypt the parameters $\{NEWC_i, NEWG_i, T_{km}, SID_l\}$ using the session key $SUK_{im}^*$ as $F_i = E_{SUK_{im}^*}(NEWC_i, NEWG_i, T_{km}, SID_l)$. $U_i$ calculates $C_i = h(N_1||TS_3||D_i')$ and send message $\{F_i, C_i, TS_3\}$ to the cloud server.

7) Cloud server verifies the validity of the timestamp as $|TS_{curr} - TS_3| \leq \Delta T$. Further, it verifies the user by computing $C_i' = h(N_1^*||TS_3||D_i'')$ and checks if the condition $C_i' \stackrel{?}{=} C_i$ holds true. If it doesn't, the server terminates the session. Else, $SM$ decrypts $F_i$ using the session key $SUK_m$. It stores temporarily the $SID_l, T_{km}$ for the session. $SM$ checks if the condition $NEWG_i \neq CURRG_i$ holds true. If it does, the cloud server deploys the gateway registration phase. Else it check if the condition $NEWC_i > CURRC_i$ hold true. If it does, the server requests the user and deploys the sensor registration phase. Otherwise, it deploys the authentication and key agreement phase.

### D. Gateway registration phase

In this phase, the $GW$ is registered by the user after successful validation of the condition $NEWG_i \neq CURRG_i$. The gateway needs to be registered before initiating the sensor registration phase. The gateway registration phase is summarized in Table IV.

1) $GW$ selects a random number $N_6$ and computes $G_k = h(GID_k||N_6)$. It send a request $\{G_k, N_6\}$ to the user through secure channel.
2) $U_i$ computes $G_k^* = h(GID_k||N_6)$ by inputting the $GID_k$ known to him/her. If the condition $G_k^* \stackrel{?}{=} G_k$ doesn't hold true, it abort the session. Otherwise, the user selects a random number $N_7$ and send message $\{ID_i, T_{km}, N_7\}$ to the gateway over secure channel.
3) $GW$ computes $VID_k = h(ID_i) \oplus GID_k \oplus x$, x being the shared secret key between the gateway and server, $SP = ID_i \oplus N_7$ , and $LK_{sm} = h(TS_4||GID_k||ID_i)$. It encrypt the parameters $\{x, VID_k, LK_{sm}, SP\}$ with key $T_{km}$ computed during the login phase and store $\{LK_{sm}, x, SP\}$ corresponding to computed $SP$ in its memory. $GW$ send $\{TE_k, TS_4\}$ to $SM$.

TABLE IV
GATEWAY REGISTRATION PHASE

| Gateway($GW$) | | User ($U_i$) | | Cloud Server($SM$) |
|---|---|---|---|---|
| Select a random number $N_6$ | | | | |
| $G_k = h(GID_k\|\|N_6)$ | | | | |
| | $\dashrightarrow$ | $\{G_k, N_6\}$ | | |
| | | | $G_k^* = h(GID_k\|\|N_6)$ | |
| | | | Verify ($G_k^* \stackrel{?}{=} G_k$) | |
| | | | If false, Session Abort | |
| | | | Else, Select a random number $N_7$ | |
| | $\dashleftarrow$ | $\{ID_i, T_{km}, N_7\}$ | | |
| $VID_k = h(ID_i) \oplus GID_k \oplus x$ | | | | |
| Choose a current timestamp $TS_4$ | | | | |
| $LK_{sm} = h(TS_4\|\|GID_k\|\|ID_i)$ | | | | |
| $SP = ID_i \oplus N_7$ | | | | |
| $TE_k = E_{T_{km}}(x, VID_k, LK_{sm}, SP)$ | | | | |
| Store $LK_{sm}, x, ID_i$ corresponding to $SP$ | | | | |
| | | $\xrightarrow{\{TE_k, TS_4\}}$ | | |
| | | | | $\|Ts_{curr} - TS_4\| \leq \Delta T$ |
| | | | | $D_{T_{km}}(TE_k) = (x, VID_k, LK_{sm}, SP)$ |
| | | | | $GID_k^* = VID_k \oplus h(ID_i) \oplus x$ |
| | | | | $Lk_{sm}^* = h(TS_4\|\|GID_k^*\|\|ID_i)$ |
| | | | | if($Lk_{sm}^* == Lk_{sm}$) then |
| | | | | $CURRG_i = GID_k^*$ |
| | | | | Store $GID_k^*, LK_{sm}, x, SP$ corresponding to $ID_i$ |
| | | | | Request user for Sensor registration phase |

4) Cloud server verifies the authenticity of the timestamp $TS_4$ as $|Ts_{curr} - TS_4| \leq \Delta T$. If the condition is not satisfied, the gateway registration process is terminated. Otherwise, $SM$ decrypts $TE_k$ using the key $T_{km}$. Then, it computes $GID_k^* = VID_k \oplus h(ID_i) \oplus x$, $LK_{sm}^* = h(TS_4||GID_k^*||ID_i)$ and checks if $G_k^* \stackrel{?}{=} G_k$ is valid.

On successful validation it updates $CURRG_i = GID_k^*$. Further it store $\{GID_k^*, LK_{sm}, x, SP\}$ in the data base corresponding to the logged-in user id tuple. Then, $SM$ send a request message to the logged-in user for sensor registration phase.

### *E. Sensor registration phase*

After successful registration of the gateway, the gateway needs to store the sensor identities in its memory to validate the sensors and communicate with them. The sensor registration phase is summarized in Table V.

TABLE V
SENSOR REGISTRATION PHASE

| User($U_i$) | Gateway($GW$) | Cloud Server($SM$) | Sensor ($S_l$) |
|---|---|---|---|
| Input $SID_l, ID_i$ | | | |
| $\dashrightarrow$ $\{SID_l, ID_i\}$ | | | |
| | $FSID_l = SID_l \oplus w_k \oplus ID_i$ | | |
| | Check for $FSID_l$ in memory | | |
| | if($FSID_l$ is present) | | |
| | Abort Session, Otherwise | | |
| | Select a random number $N_8$ | | |
| | $SG = N_8 \oplus h(w_k \| GID_k \| ID_i)$ | | |
| | Store $FSID_l, SG$ | | |
| | $V_{gc} = GID_k \oplus h(x \| TS_5)$ | | |
| | $\dashrightarrow \{SG\}$ | | |
| | | | Store $SG$ |
| | $\{V_{gc}, TS_5\} \longrightarrow$ | | |
| | | $\|TS_{curr} - TS_5\| \leq \Delta T$ | |
| | | $V_{gc}^* = GID_K^* \oplus h(x \| TS_5)$ | |
| | | if($V_{gc}^* == V_{gc}$) | |
| | | $CURRC_i = CURRC_i + 1$ | |
| | | if($NEWC_i > CURRC_i$) | |
| | | Request user for sensor registration phase. | |
| | | else | |
| | | if($NEWC_i == CURRC_i$) | |
| | | Authentication and key agreement phase | |

1) $U_i$ inputs $SID_l, ID_i$, and send it to the gateway through a secure channel.
2) $GW$ computes $FSID_l = SID_l \oplus w_k \oplus ID_i$ and check the presence of sensor in its memory by comparing computed $FSID_l$ with the stored one. If it exist then, gateway aborts the session otherwise it select a random number $N_8$ and compute $SG = N_8 \oplus h(w_k || GID_k || ID_i)$. Computed parameters $FSID_l, SG$ are stored in the memory and $V_{gc}$ is computed. $GW$ send $SG$ over a secure channel to the sensor and $\{V_{gc}, TS_5\}$ to cloud server over public channel. $SID_l$ store $SG$ in its memory.
3) $SM$ computes $V_{gc}^* = GID_K^* \oplus h(x || TS_5)$. If the condition $V_{gc}^* \stackrel{?}{=} V_{gc}$ doesn't hold true, $SM$ terminates the session. Else, it computes $CURRC_i = CURRC_i + 1$ and continue this phase until $NEWC_i > CURRC_i$. If the condition $CURRC_i == NEWC_i$ satisfies it deploys the authentication and key agreement phase.

### *F. Authentication and key agreement phase*

This phase ensures all the entities authenticate to each other, and only legitimate entities can access the resources of the system. This phase is summarized in Table VI.

1) $SM$ selects random number $N_9$ and computes $N_7^* = SP \oplus ID_i$, $M2 = LK_{sm} \oplus GID_k^* \oplus h(N_7^*)$, $M3 = h(N_9 || ID_i || GID_k^* || x)$. It encrypts the parameter $\{N_9, SID_l, M3\}$ and send $\{M2, M4, TS_6, SP\}$ to $GW$.
2) $GW$ receives the message and verifies the validity of timestamp $TS_6$ as $|TS_{curr} - TS_6| \leq \triangle T$. On success it search for $ID_i$ corresponding to received $SP$ . If not found it aborts the session else computes $N_7^{**} = SP \oplus ID_i$, and $LK_{sm}^* = M2 \oplus GID_k \oplus h(N_7^{**})$. It decrypt $M4$ as $A_1 = D_{LK_{sm}}(M4) = (N_9, SID_l, M3)$ and computes $M3^* = h(N_9 || ID_i || GID_k || x)$. Further if $M3^* \stackrel{?}{=} M3^*$ doesn't satisfy, $GW$ terminate the session else it computes $FSID_l^* = SID_l \oplus w_k \oplus ID_i$ and check for its presence in memory. If not found then, it aborts the session else it computes $TFID_l = SID_l \oplus SG$, $M5 = h(TFID_l || TS_7)$, and $M6 = N_{10} \oplus h(SID_l || TFID_l)$. It send $\{M5, M6, TS_7\}$ to $SID_l$.
3) $S_l$ verifies the timestamp as $|TS_{curr} - TS_7| \leq \triangle T$. If the verification is unsuccessful the session is terminated. Else, it computes $TFID_l^* = SID_l \oplus SG$, $M5^* = h(TFID^* || TS_7)$. If the condition $M5 \stackrel{?}{=} M5^*$ does not hold true, $S_l$ terminates the session. Otherwise it computes $N_{10}^* = M6 \oplus h(SID_l || TFID_l^*)$, $M7 = N_{11} \oplus TFID^*$, session key $SK_1 = h(N_{11} || N_{10}^* || TFID_l^*)$ and $M8 = h(SID_l || N_{10}^* || N_{11} || TS_8)$. Then $S_l$ send $\{M7, M8, TS_8\}$ to $GW$.
4) $GW$ verifies the timestamp as $|TS_{curr} - TS_8| \leq \triangle T$. On successful verification it computes $N_{11}^* = M7 \oplus TFID_l$, $M8^* = h(SID_l || N_{10} || N_{11}^* || TS_8)$ and session key $SK_1^*$. Then it checks if the condition $M8 \stackrel{?}{=} M8^*$ doesn't hold true, $GW$ terminates the session. Else, sensor is authenticated by $GW$ and it further computes $M9 = N_{12} \oplus h(SID_l || GID_k)$, session key $SK_2 = h(N_9 || N_{12} || GID_k || SID_l)$ and $M10 = h(SID_l || x || N_{12})$ to check the authenticity of itself with the cloud server. It sends $\{M9, TS_9\}$ to $SM$.
5) $SM$ validates the timestamp $TS_9$ and computes $N_{12}^* = M9 \oplus h(SID_l || GID_k)$, $M10^* = h(SID_l || x || N_{12}^*)$, and $SK_2^*$. If the condition $M10^* \stackrel{?}{=} M10$ holds true, then $G_k$ and $SM$ are authenticated. Else cloud server terminates the session.

## V. SECURITY VERIFICATION AND ANALYSIS

We have simulated the proposed protocol with the proverif and proved its correctness. An informal security analysis of the proposed protocol is also discussed against some known attacks.

### *A. Formal verification using Proverif*

Proverif is an autonomous tool responsible for formally analyzing security protocols. It is based on PI calculus and ensures various properties such as secrecy, authentication of the protocol. It imposes no limit on the number of concurrent execution of the protocol by the attacker [22]. It supports a variety of cryptographic primitives such as encryption/decryption, hash functions, digital signature, etc. The verification is independent of the number of messages, participants, and channels between the participants. The tool outputs a detailed trace of the intruder's attacks if the protocol is insecure. In this section, we have used this tool to formally prove the secrecy and authentication properties of our proposed protocol. The result

TABLE VI
AUTHENTICATION AND KEY AGREEMENT PHASE

| Cloud Server ($SM$) | Gateway($GW$) | Sensors($S_l$) |
|---|---|---|
| Choose random number $N_9$ | | |
| $N_7^* = SP \oplus ID_i$ | | |
| $M2 = LK_{sm} \oplus GID_k^* \oplus h(N_7^*)$ | | |
| $M3 = h(N_9 \| ID_i \| GID_k^* \| x)$ | | |
| $M4 = E_{Lk_{sm}}(N_9, SID_l, M3)$ | | |
| | $\xrightarrow{\{M2, M4, TS_6, SP\}}$ | |
| | $\|TS_{curr} - TS_6\| \leq \triangle T$ | |
| | Search for $ID_i$ corresponding to $SP$ | |
| | $N_7^{**} = SP \oplus ID_i$ | |
| | $LK_{sm}^* = M2 \oplus GID_k \oplus h(N_7^{**})$ | |
| | $A_1 = D_{LK_{sm}^*}(M4) = (N_9, SID_l, M3)$ | |
| | $M3^* = h(N_9 \| ID_i \| GID_k \| x)$ | |
| | if($M3 == M3^*$)then | |
| | $FSID_l^* = SID_l \oplus w_k \oplus ID_i$ | |
| | Check for $FSID_l^*$ in memory | |
| | if($FSID_l^*$ is present) | |
| | Choose a timestamp $TS_7$ | |
| | $TFID_l = SID_l \oplus SG$ | |
| | $M5 = h(TFID_l \| TS_7)$ | |
| | Select a random number $N_{10}$ | |
| | $M6 = N_{10} \oplus h(SID_l \| TFID_l)$ | |
| | Else, Abort session | |
| | | $\xrightarrow{\{M5, M6, TS_7\}}$ |
| | | $\|TS_{curr} - TS_7\| \leq \triangle T$ |
| | | $TFID_l^* = SID_l \oplus SG$ |
| | | $M5^* = h(TFID_l^* \| TS_7)$ |
| | | if($M5^* == M5$) |
| | | $N_{10}^* = M6 \oplus h(SID_l \| TFID_l^*)$ |
| | | Select a random number $N_{11}$ |
| | | $M7 = N_{11} \oplus TFID_l^*$ |
| | | $SK_1 = h(N_{11} \| N_{10}^* \| TFID_l^*)$ |
| | | Choose the current timestamp $TS_8$ |
| | | $M8 = h(SID_l \| N_{10}^* \| N_{11} \| TS_8 \| SK_1)$ |
| | $\xleftarrow{\{M7, M8, TS_8\}}$ | |
| | $\|TS_{curr} - TS_8\| \leq \triangle T$ | |
| | $N_{11}^* = M7 \oplus TFID_l$ | |
| | $SK_1^* = h(N_{11}^* \| N_{10} \| TFID_l)$ | |
| | $M8^* = h(SID_l \| N_{10} \| N_{11}^* \| TS_8 \| SK_1^*)$ | |
| | if($M8^* == M8$) | |
| | Accept the Sensor | |
| | Choose the current timestamp $TS_9$ | |
| | Select a random number $N_{12}$ | |
| | $M9 = N_{12} \oplus h(SID_l \| GID_k)$ | |
| | $SK_2 = h(N_9 \| N_{12} \| GID_k \| SID_l)$ | |
| | $M10 = h(SID_l \| x \| N_{12} \| SK_2)$ | |
| $\xleftarrow{\{M9, M10, TS_9\}}$ | | |
| $\|TS_{curr} - TS_9\| \leq \triangle T$ | | |
| $N_{12}^* = M9 \oplus h(SID_l \| GID_k)$ | | |
| $SK_2^* = h(N_9 \| N_{12}^* \| GID_k \| SID_l)$ | | |
| $M10^* = h(SID_l \| x \| N_{12}^* \| SK_2^*)$ | | |
| if($M10^* == M10$) | | |
| Accept the gateway and Sensors | | |

of analyzing the proposed protocol using Proverif is shown in Fig 3. The result shows that the proposed protocol resists passive and active attacks. The session keys in the proposed protocol are also not compromised.

```
asit@hp:~/Desktop/Proverif/proverif2.00$ ./proverif myprotocol.pv |grep "RES"
RESULT not attacker(sessionkey[]) is true.
RESULT not attacker(id[]) is true.
RESULT not attacker(gid[]) is true.
RESULT not attacker(sid[]) is true.
RESULT not attacker(Fk[]) is true.
RESULT not attacker(lm[]) is true.
RESULT inj-event(enduser(id_83,pwd_85,bio_86,gid_87,sid_84,NewG_88,Newc_89)) ==>
 inj-event(beginuser(id_83,pwd_85,bio_86,gid_87,sid_84,NewG_88,Newc_89)) is true
.
RESULT inj-event(endserver) ==> inj-event(beginserver) is true.
RESULT inj-event(endgateway(gid_91)) ==> inj-event(begingateway(gid_91)) is true
.
RESULT inj-event(endsensor(sid_92)) ==> inj-event(beginsensor(sid_92)) is true.
asit@hp:~/Desktop/Proverif/proverif2.00$
```

Fig. 3. Analysis of proposed protocol using proverif tool

## B. Informal security analysis

This subsection discusses the security features of our scheme, and our scheme resists most of the known attacks listed as follows:

*1) User anonymity:* The proposed scheme provides user anonymity. All communication between parties takes place on the public channel. In the proposed protocol, during the login phase, even if the adversary intercepts the login message $\{LINF_i, TEM_i, TS_1\}$, it can't derive the identity of the user from it as the parameters are encrypted using the secret key.

*2) Gateway identity guessing attack:* The gateway identity $GID_k$ is masked in various ways in each phase. No registration or authentication message includes the $GID_k$ directly. Even if the adversary can intercept the message $\{TE_k, TS_4\}$, it still fails to get the gateway id $GID_k$ as it is encrypted using session key $T_{km}$. Thus, the proposed scheme is resistant to gateway guessing attacks.

*3) Sensor guessing attack:* During the authentication phase, even though the adversary intercepts the message $\{M5, M6, TS_7\}$, $\{M7, M8, TS_8\}$, it still fails to guess the sensor id $SID_l$ as they are hashed with other parameters.

*4) Replay attack:* Suppose the attacker eavesdrops messages during the authentication and key agreement phase over the public channel. It still can not replay the previous login or authentication messages because the proposed scheme uses a combination of timestamp and generation of random numbers at each session. $\triangle T$ is kept sufficiently small to ensure no replay attacks.

*5) Perfect forward secrecy:* The proposed scheme provides perfect forward secrecy. During the login phase, even if the adversary obtains the secret key $RK_{im}^*$, he/she will still not be able to compute the session keys. It needs to compute $D_i'$, which is not computationally feasible.

*6) Password guessing attack:* The proposed scheme is resistant to a password guessing attack. Assume that the adversary steals or finds a user's smart card and retrieve $\{B_i, E_i, N_1, H_{b_i}\}$ from it. He/she will still not able to compute the password of users as it is secured using the one-way hash function.

*7) Mutual authentication:* The user and server mutually authenticate themselves during the login phase. The server authenticates the user by searching $TEM_i'$ from its database. On the other hand, the server is authenticated by the user using $VSU_{im}$. Similarly, during the authentication and key agreement phase, the cloud server and gateway mutually authenticate themselves by using $M3$, $M10$. Further, the gateway and sensors mutually authenticate themselves using $M5$ and $M8$. This shows that mutual authentication between all the entities is achieved in our proposed scheme.

*8) Session key disclosure attack:* The proposed scheme resists the session key disclosure attack. All the session keys are generated by masking it with a random number. It is very difficult for the attacker to generate the same random number again. No login, authentication, registration messages include the session key directly.

## C. Comparative analysis

This section demonstrates the comparison of computational and communicational cost among the proposed scheme and other related schemes. We denote hash function, encryption/decryption, ECC point as $T_h, T_{em}, T_{ecm}$ respectively. The experiments are carried out on a PC running Windows 10, with an Intel(R) Core(TM) i7 4th generation CPU @3.40 GHz having 4 cores and 4 GB RAM. The time taken by $T_h, T_{em}, T_{ecm}$ are 0.0009s, 0.0010, and 0.0253s respectively. A comparative analysis of the proposed scheme with other schemes based on

the computational cost is given in Table VII, which shows that the proposed scheme is less than the other schemes. For the communicational cost, we have taken hash function, encryption/decryption, ECC point as 256 bit, 128 bit, 160 bit, respectively. Besides, identity/password/random number/time stamp is taken as 32 bits. Table VII demonstrates that the proposed scheme has less computational and communicational cost compared to other related schemes.

TABLE VII
COMPARATIVE COST ANALYSIS

| Scheme⟶ Cost↓ | [23] | [24] | [25] | [26] | Proposed Scheme |
|---|---|---|---|---|---|
| Computational cost | $12T_h + 3T_{ecm}$ | $13T_h + 4T_{em}$ | $20T_h$ | $16T_h + 3T_{em}$ | $17T_h + 2T_{em}$ |
| Execution time | $\approx$ 0.0867s | $\approx$ 0.0157s | $\approx$ 0.018s | $\approx$ 0.0174s | $\approx$ 0.0173s |
| Communication cost | 1632 bits | 1920 bits | 1280 bits | 1920bits | 1792bits |

## VI. CONCLUSION

IoT is a trend that is unlikely to fade anytime soon, and designing lightweight cryptographic schemes suitable for IoT deployment remains a research challenge. We have developed the network model for the IoT system, which can be used for small scale applications. The proposed scheme uses lightweight operations such as hash, XOR, along with ECC, and AES for better security and key generation. The protocol supports interoperability and is feasible to deploy in low resource-constrained network devices. The proposed scheme is verified using Proverif. Further, an informal security analysis of the proposed scheme demonstrates that it can resist most of the known attacks. In addition, the comparison study based on computational cost shows that the proposed scheme takes less time than its counterparts. In the future, we intend to modify the proposed protocol for a multi gateway environment to support large scale applications. We aim to check the practical applicability of the scheme by simulating it in NS2.

## VII. ACKNOWLEDGMENT

This work is carried out under Information Security Education and Awareness (ISEA) Phase II, funded by Ministry of Electronics and Information Technology (MeitY).

## REFERENCES

[1] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Comput. Netw.*, 54(15):2787–2805, oct 2010.

[2] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani. Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges. *IEEE Wireless Communications*, 24(3):10–16, June 2017.

[3] Clare Hopping. Iot malware tripled in the first half of 2018. In *https://www.itpro.co.uk/malware/31945/iot-malware-tripled-in-the-first-half-of-2018 [Online]*, september 2019.

[4] M. M. Hossain, M. Fotouhi, and R. Hasan. Towards an analysis of security issues, challenges, and open problems in the internet of things. In *2015 IEEE World Congress on Services*, pages 21–28, June 2015.

[5] Bojan M Bakmaz Zoran S Bojkovic and Miodrag R Bakmaz. Security issues in wireless sensor networks. 01 2008.

[6] Anne Geraci. *IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries*. IEEE Press, Piscataway, NJ, USA, 1991.

[7] G. Aloi, G. Caliciuri, G. Fortino, R. Gravina, P. Pace, W. Russo, and C. Savaglio. Enabling iot interoperability through opportunistic smartphone-based mobile gateways. *Journal of Network and Computer Applications*, 81:74 – 84, 2017.

[8] Ming-Chin Chuang and Meng Chang Chen. An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics. *Expert Systems with Applications*, 41(4):1411–1418, 2014.

[9] Ruhul Amin, Neeraj Kumar, GP Biswas, Rahat Iqbal, and Victor Chang. A light weight authentication protocol for iot-enabled devices in distributed cloud computing environment. *Future Generation Computer Systems*, 78:1005–1019, 2016.

[10] Sheetal Kalra and Sandeep K Sood. Secure authentication scheme for iot and cloud servers. *Pervasive and Mobile Computing*, 24:210–223, 2015.

[11] Saru Kumari, Marimuthu Karuppiah, Ashok Kumar Das, Xiong Li, Fan Wu, and Neeraj Kumar. A secure authentication scheme based on elliptic curve cryptography for iot and cloud servers. *The Journal of Supercomputing*, 74(12):6428–6453, 2018.

[12] Chia-Hui Liu and Yu-Fang Chung. Secure user authentication scheme for wireless healthcare sensor networks. *Computers & Electrical Engineering*, 59:250–261, 2017.

[13] Sravani Challa, Ashok Kumar Das, Vanga Odelu, Neeraj Kumar, Saru Kumari, Muhammad Khurram Khan, and Athanasios V Vasilakos. An efficient ecc-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks. *Computers & Electrical Engineering*, 69:534–554, 2018.

[14] Mohammad Sabzinejad Farash, Muhamed Turkanović, Saru Kumari, and Marko Hölbl. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment. *Ad Hoc Networks*, 36:152–176, 2016.

[15] Ruhul Amin, SK Hafizul Islam, GP Biswas, Muhammad Khurram Khan, Lu Leng, and Neeraj Kumar. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. *Computer Networks*, 101:42–62, 2016.

[16] Arezou Ostad-Sharif, Hamed Arshad, Morteza Nikooghadam, and Dariush Abbasinezhad-Mood. Three party secure data transmission in iot networks through design of a lightweight authenticated key agreement scheme. *Future Generation Computer Systems*, 100:882–892, 2019.

[17] Parwinder Kaur Dhillon and Sheetal Kalra. A lightweight biometrics based remote user authentication scheme for iot services. *Journal of Information Security and Applications*, 34:255–270, 2017.

[18] Prosanta Gope and Biplab Sikdar. Lightweight and privacy-preserving two-factor authentication scheme for iot devices. *IEEE Internet of Things Journal*, 6(1):580–589, 2018.

[19] Yo-Hsuan Chuang, Nai-Wei Lo, Cheng-Ying Yang, and Ssu-Wei Tang. A lightweight continuous authentication protocol for the internet of things. *Sensors*, 18(4):1104, 2018.

[20] Lu Zhou, Xiong Li, Kuo-Hui Yeh, Chunhua Su, and Wayne Chiu. Lightweight iot-based authentication scheme in cloud computing circumstance. *Future Generation Computer Systems*, 91:244–251, 2019.

[21] Geeta Sharma and Sheetal Kalra. Advanced lightweight multi-factor remote user authentication scheme for cloud-iot applications. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–24, 2019.

[22] HMN Al Hamadi, CY Yeun, MJ Zemerly, MA Al-Qutayri, and A Gawanmeh. Verifying mutual authentication for the dlk protocol using proverif tool. *International Journal for Information Security Research*, 2(1):256–265, 2012.

[23] Jaewook Jung, Jongho Moon, Donghoon Lee, and Dongho Won. Efficient and security enhanced anonymous authentication with key agreement scheme in wireless sensor networks. *Sensors*, 17(3):644, 2017.

[24] Sooyeon Shin and Taekyoung Kwon. A lightweight three-factor authentication and key agreement scheme in wireless sensor networks for smart homes. *Sensors*, 19(9):2012, 2019.

[25] Q. Lyu, N. Zheng, H. Liu, C. Gao, S. Chen, and J. Liu. Remotely access "my" smart home in private: An anti-tracking authentication and key agreement scheme. *IEEE Access*, 7:41835–41851, 2019.

[26] Fan Wu, Lili Xu, Saru Kumari, and Xiong Li. An improved and anonymous two-factor authentication protocol for health-care applications with wireless medical sensor networks. *Multimedia Systems*, 23(2):195–205, 2017.