PAKIT: Proactive Authentication and Key Agreement Protocol for Internet of Things

Seyed Farhad Aghili*, Hamid Mala*, Mohammad Shojafar† and Mauro Conti†
*Department of Information Technology Engineering, University of Isfahan, Isfahan, Iran
(e-mail: {sf.aghili, h.mala}@eng.ui.ac.ir)
†Department of Mathematics, University of Padua, Padua, Italy
(e-mail: {shojafar, conti}@math.unipd.it)

***Abstract*—Internet of Things (IoT) holds great promise for many life-improving applications like health-care systems. In IoT systems, providing secure authentication and key agreement scheme that considers compromised entities is an important issue. State-of-the-arts tackle this problem, but they fail to address compromised entity attack and have high computation cost. Motivated by these considerations, in this paper, we propose an energy-efficient proactive authentication and key agreement scheme called PAKIT for IoT systems. The security of PAKIT scheme is validated using the ProVerif tool. Moreover, the efficiency of PAKIT is compared with the predecessor schemes proposed for IoT systems. The results of the experiments show that PAKIT is efficient and suitable for real-world IoT applications by utilizing lightweight functions, such as hash and XOR.**

***Index Terms*—Internet of thing (IoT), authentication, secret sharing, compromised entity.**

I. Introduction

The Internet of Things (IoT) systems have many life-improving applications such as health-care, transportation, smart grid and so on. In IoT systems, sensors are responsible for gathering objects' information, and the cluster heads and gateways are responsible for the transmission of collected data from the sensors to the corresponding application systems controlled by the users. RFID and WSNs are the examples of sensors layer of the IoT to collect data from physical objects.

Using mobile entities in IoT systems provides attackers the opportunity to compromise entities and impersonate legal entities to access the vital information. For example, in health-care systems, if an adversary accesses to a patient's data, s/he can change this crucial information and induce the fake information to a doctor, cause a wrong prescription and kill the patient. Besides, designed schemes should be effective for recourse-constrained entities such as sensors implemented in the patient's body. Thus, designing an energy-efficient authentication and key agreement scheme which does not only provide security and privacy between entities but also consider compromised entities threat for IoT systems is indispensable.

A comprehensive study for the authentication protocols in the IoT system is presented in [1]. In most of them, authors employ schemes that are designed for use in traditional systems such as Wireless Sensor Networks (WSNs) which do not consider compromised entity and cannot provide complete security and privacy, if they are adapted to the IoT systems directly.

Some methods mainly address various types of attacks, such as replay attack [2], privileged-insider attack [3], user impersonation attack [4], password guessing attack [5]. Some other works proposed for IoT systems do not consider or are not robust against entity compromised attack [6]–[8]. In particular, in recent years, several security schemes have been proposed in the mobile internet domain (e.g., [9]–[11]) to solve the entity compromised problem.

In [9], the authors present a secure key management and user authentication scheme named SAKA-FC which is classified as a fog-supported secure communication protocol which employs one-way hash function and bitwise XOR which are supportable by resource-constrained IoT devices. The scheme suffers from controlling the privileged-insider attack and cannot provide a secure environment for compromising entity attack. Interestingly, our presented lightweight protocol is secure against entity compromised attack using secret sharing and can avoid the limitation raised in [9].

In [10], the authors present a new secure authentication protocol using secure key agreement and private key security mechanisms mutually applied for two-party computation for a mobile Internet environment. For the validation, they tend to provide robust security analysis to prove that the proposed protocol can achieve all known security requirements of authentication protocols for a mobile Internet environment. Although the paper is of interest, it has two main drawbacks compared to our proposed method. First, their method is two-tier entity type approach (user and server). In contrast, our method is a four-tier protocol in which we add partial private key applied in registration phase between the user and gateway that supports the security of the channel. Second, the scheme in [10] is not a proactive secret sharing scheme and an attacker can compromise sufficient number of shares (equal to a predefined threshold) and make their scheme vulnerable against entity compromise attack. Conversely, our method uses a proactive Shamir's secret sharing [12] in which entities update their shares after each successful session such that an adversary cannot use compromised shares to do entity compromise attack.

Furthermore, the authors in [11] introduce an ultra-lightweight mutual-authentication protocol based on Shamir's secret sharing. Their scheme offers a robust defense against a broad range of typical attacks. It provides double verification, authentication, and dynamic update on the IoT system. In detail, they introduce overtime-exit function using a session control mechanism to regulate the round-trip time of every challenge-response cycle. The main drawback of the method is that it requires a trusted party to hold the complete secret key after it is recovered while in our proposed method the trusted party does not need to store it.

In this paper, several questions arise, like: is it possible to propose a secure protocol that can cover compromised entity challenges among each other? How to model a lightweight proactive authentication and key agreement scheme to satisfy the security and privacy in IoT systems? The answer to these questions is the goal of the paper.

A. Contribution

We will introduce a dynamic and computationally efficient proactive authentication and key agreement scheme called PAKIT in which we provide the security and privacy in IoT systems. Hence, the main contributions are as follows:

- We propose PAKIT scheme for IoT systems which is lightweight for the resource-constrained sensors;
- PAKIT scheme is secure against entity compromised attack using secret sharing;
- In PAKIT scheme, the shared value of each legal entity is updated, which makes it proactive;
- We scrutinize the security of PAKIT scheme from formal and informal (ProVerif tool [13]) point of view;
- Finally, the efficiency of PAKIT is compared to predecessor schemes. Due to the presented results, PAKIT can be adapted for resource-constrained sensors in IoT.

B. Roadmap

The remainder of the paper is organized as follows: Section II provides models and related architecture. Section III reports the proposed lightweight authentication protocol (called PAKIT) for IoT systems. Next, the informal and formal security analysis and performance evaluation of the proposed protocol are presented in Section IV. Finally, Section V concludes the paper and presents future directions of research.

II. Preliminary

This section is followed by a presentation of the threat model and security assumptions, description of our proposed architecture and the definition of the secure secret sharing. The notation used in this paper is also presented in Table I.

TABLE I: Notation.

Notation	Description
SN, GW, CH	The sensor node, gateway and cluster head
ID_u, ID_g, ID_c, ID_s	Identity of user, GW, CH, SN, respectively
r_u, r_g, r_c, r_s	Random numbers generated by user, GW, CH, SN, respectively
p	A large prime integer
G, g	A multiplicative group and its generator, respectively
M_u, M_c, M_s	The masked identity of the user, CH and SN
S	The secret key , where $S \in Z_p^*$
s	The master key of the GW
SK_u, SK_g,SK_c, SK_s	The session key calculated respectively by the user, GW, CH and SN
T_i	The i-th time-stamp
$h(\cdot)$	A one-way hash function
$\oplus$, $\|$	XOR and concatenation operations

A. Threat Model

The threat model for our proposed scheme in IoT systems is based on two main assumptions. i) the model proposed by the Dolev-Yao [14] which the adversary can intercept all the transferred messages in the protocol and also can modify, delete and block messages. ii) The adversary can also compromise a gateway, cluster head, and sensor nodes, and obtain all information stored in their database.

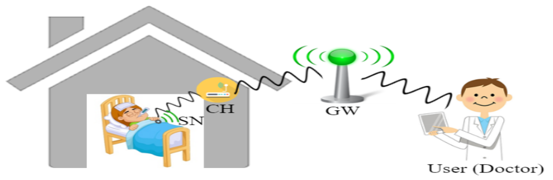

Fig. 1: The architecture of our proposed scheme.

B. Proposed Architecture

Our architecture is comprised of four main entities as shown in Fig. 1. The first entity is a gateway GW which is responsible for executing the initialization phase of the scheme and also should provide the secret shares that the other entities requested. The second entity is a user registered to the GW and can login to the system by some delivered security parameters from GW and her/his identity (e.g., user's identification string, user's biometric template) and password. The user can use smart-card or smartphone to pass the login phase. The next entity is the sensor node SN which collects the environmental data and delivers them to the legal user –note that in IoT systems there are several numbers of sensor nodes with the specified identity, here only one of them is considered with an identity ID_s. The last entities are cluster heads CH. These entities could be the base stations, mobile sets and so on. Each cluster head is responsible for providing the session key for the user, gateway and sensor node whenever they want to establish a new session. In the proposed architecture, all communication channels between these four entities are the insecure channel and all entities, except the user, could be compromised.

C. Secret Sharing Scheme

In a (t, w)-*threshold* secret sharing scheme, a secret S is divided into w partial information, called shares, and each entity receives one piece such that following security properties are met.

- Any t or more entities can reconstruct S.
- Fewer than t entities get nothing on S.

In PAKIT, we use Shamir's (t, w)-*threshold* secret sharing scheme proposed in 1979 [12]. Shamir's (t, w)-*threshold* secret sharing scheme is based on interpolated polynomial and has three phases as below.

- *Initialization phase* The special entity (GW in our scheme) chooses $S \in Z_p^*$ as a secret and chooses w distinct non-zero elements of $x_i \in Z_p$, $1 \leq i \leq w$. Then, GW keeps x_1 and sends the remained elements to the $w - 1$ entities.
- *Share distribution phase* Now, GW chooses $t - 1$ elements $a_j \in Z_p$, $1 \leq j \leq t - 1$ and computes

$y_i = a(x_i) = S + \sum_{j=1}^{t-1} a_j x_i^j \ mod\ p$. Eventually, GW gives $v_i = x_i \| y_i$ to the $i\text{-}th$ entity. GW also keeps one share for itself.

- $Secret\ reconstruction\ phase$ At this point, a group of at least t entities can reconstruct S by executing the Lagrange interpolation equation.
  $S' = f(0) = \sum_{i=1}^{t} y_i \prod_{k=1,k\neq i}^{t} \frac{x_k}{x_k - x_i}\ mod\ p$

D. Discrete Logarithm (DL) Problem

Let p be a large prime integer, and G be a multiplicative group with generator g. Given an element $m \in G$ calculating $S \in Z_p^*$ such that $m = g^S\ mod\ p$ is called the DL problem.

III. The PAKIT Scheme

Our proposed PAKIT scheme consists of three important phases: i) Setup phase; ii) Registration phase; iii) Authentication and key agreement phase. Details are provided below.

A. Setup Phase

In this phase, the gateway GW runs Shamir's $(4,4)$-$threshold$ secret sharing scheme as described in Section II and securely sends $v_u = x_u \| y_u$, $v_c = x_c \| y_c$, and $v_s = x_s \| y_s$ to the user, CH, and SN, respectively. GW holds the first share $v_g = x_g \| y_g$ and $g^S\ mod\ p$ and sends p and g to CH.

B. Registration Phase

In this phase, user, CH, and SN generate the random number r_u, r_c and r_s respectively. Then they compute the masked identities $M_u = h(ID_u \| r_u)$, $M_c = h(ID_c \| r_c)$, and $M_s = h(ID_s \| r_s)$, independently, and securely send them to the GW. Finally, user, CH, and SN receive $f_u = h(M_u \| s)$, $f_c = h(M_c \| s)$, and $f_s = h(M_s \| s)$, respectively, in which s is the master key of GW.

C. Authentication and Key Agreement Phase

The goals of this phase of PAKIT scheme are i) providing mutual authentication between all entities; ii) establishing a new session key between all entities; iii) updating all the shares of constant secret (proactive secret sharing). Steps of this phase of the protocol are described as below:

Algorithm 1 Step 1: $User \rightarrow GW$

```
Input: $v_u = x_u \| y_u$, $f_u$, $M_u$, $M_s$
Output: $Msg_1$
1: Generates random number $k_u$, Time-stamp $T_1$
2: $M_1 \leftarrow k_u \oplus h(M_u \| f_u \| T_1)$, $M_2 \leftarrow M_s \oplus h(f_u \| T_1)$
3: $M_3 \leftarrow v_u \oplus h(k_u \| T_1)$, $M_4 \leftarrow h(M_u \| k_u \| f_u \| T_1 \| M_s \| M_3)$
4: return $Msg_1 = M_u, M_1, M_2, M_3, M_4, T_1$
```

$Step$ 1 First, the user runs Algorithm 1 and provides Msg_1 and sends it to the gateway GW through an insecure channel. This message involves the current share (v_u) of the user. The time-stamp T_1 is used to prevent a replay attack.

$Step$ 2 Upon receiving the message Msg_1, GW executes Algorithm 2 and if all the conditions are satisfied, the message Msg_2 is computed and sent to CH. Besides, GW authenticates the user as a legitimate one who can access the sensor nodes information. In this algorithm, v_g is also the GW's share transferred to CH in ciphertext form.

$Step$ 3 CH runs Algorithm 3, by checking whether the time-stamp T_2 and M_9 are valid. If no, the authentication fails. Otherwise, CH authenticates GW and computes M_{11} and M_{12} and sends Msg_3 to the sensor node SN. In this message, T_3 is the current CH's time-stamp.

Algorithm 2 Step 2: $GW \rightarrow CH$

```
Input: $v_g = x_g \| y_g$, $s$, $M_c$, $Msg_1$
Output: $Msg_2$
1: Generates Time-stamp $T_2$
2: if $|T_1 - T_2| > \Delta T$ then return $\perp$
3: else
4:     $f'_u \leftarrow h(M_u \| s)$, $k'_u \leftarrow M_1 \oplus h(M_u \| f'_u \| T_1)$
5:     $M'_s \leftarrow M_2 \oplus h(f'_u \| T_1)$, $M'_4 \leftarrow h(M_u \| k'_u \| f'_u \| T_1 \| M'_s \| M_3)$
6:     if $M'_4 \neq M_4$ then return $\perp$
7:     else
8:         User is authenticated
9:         $v'_u \leftarrow M_3 \oplus h(k'_u \| T_1)$, $f'_c \leftarrow h(M_c \| s)$, $f'_s \leftarrow h(M'_s \| s)$
10:        $M_5 \leftarrow v'_u \oplus h(M'_s \| T_2)$, $M_6 \leftarrow k'_u \oplus v'_u$
11:        $M_7 \leftarrow v_g \oplus h(f'_c \| T_2)$, $M_8 \leftarrow M'_s \oplus h(v_g \| T_2)$
12:        $M_9 \leftarrow h(v_g \| f'_c \| M_8 \| T_2 \| v'_u)$, $M_{10} \leftarrow f'_s \oplus h(M_c \| T_2 \| k'_u)$
13:        return $Msg_2 = M_5, M_6, M_7, M_8, M_9, M_{10}, T_2$
14:    end if
15: end if
```

Algorithm 3 Step 3: $CH \rightarrow SN$

```
Input: $v_c = x_c \| y_c$, $f_c$, $M_c$, $Msg_2$
Output: $Msg_3$
1: Generates Time-stamp $T_3$
2: if $|T_2 - T_3| > \Delta T$ then return $\perp$
3: else
4:     $v'_g \leftarrow M_7 \oplus h(f_c \| T_2)$, $M'_s \leftarrow M_8 \oplus h(v'_g \| T_2)$
5:     $v'_u \leftarrow M_5 \oplus h(M'_s \| T_2)$, $M'_9 \leftarrow h(v'_g \| f_c \| M_8 \| T_2 \| v'_u)$
6:     if $M'_9 \neq M_9$ then return $\perp$
7:     else
8:         GW is authenticated
9:         $k'_u \leftarrow M_6 \oplus v'_u$
10:        $M_{11} \leftarrow M_c \oplus h(M'_s \| T_3)$
11:        $M_{12} \leftarrow k'_u \oplus h(M_c \| T_3 \| T_2)$
12:        return $Msg_3 = M_{10}, M_{11}, M_{12}, T_2, T_3$
13:    end if
14: end if
```

$Step$ 4 Once the message Msg_3 is received, SN executes Algorithm 4 and checks whether T_3 and f_s are valid. If yes, CH is authenticated. Then, SN generates a new random number k_s and sends Msg_4 to the CH. In this message, the SN's share v_s is protected by M_{13}, the ciphertext M_{15} is computed for proving the integrity of M_{13} and M_{14}, and the current SN's time-stamp T_4 is used to prevent replay attack.

Algorithm 4 Step 4: $SN \rightarrow CH$

```
Input: $v_s = x_s \| y_s$, $f_s$, $M_s$, $Msg_3$
Output: $Msg_4$
1: Generates Time-stamp $T_4$
2: if $|T_3 - T_4| > \Delta T$ then return $\perp$
3: else
4:     $M'_c \leftarrow M_{11} \oplus h(M'_s \| T_3)$
5:     $k'_u \leftarrow M_{12} \oplus h(M'_c \| T_3 \| T_2)$
6:     $f'_s \leftarrow M_{10} \oplus h(M'_c \| T_2 \| k'_u)$
7:     if $f'_s \neq f_s$ then return $\perp$
8:     else
9:         CH is authenticated
10:        Generates Random number $k_s$
11:        $M_{13} \leftarrow v_s \oplus h(M_s \| M'_c \| T_4)$
12:        $M_{14} \leftarrow k_s \oplus h(M'_c \| T_4)$
13:        $M_{15} \leftarrow h(M_{13} \| M'_c \| T_4 \| k_s)$
14:        return $Msg_4 = M_{13}, M_{14}, M_{15}, T_4$
15:    end if
16: end if
```

Step 5 After receiving the message Msg_4, CH runs Algorithm 5 and if the value of T_4 is verified, it extracts the last share of the secret which is the v_s. At this point, if M_{13} is valid, CH mutually authenticates SN and employs all collected shares (i.e., $v_u = x_u \| y_u$, $v_g = x_g \| y_g$, $v_c = x_c \| y_c$ and $v_s = x_s \| y_s$) to compute the secret S' by using the equation presented in Algorithm 5 Line 9. Note that, in this equation $x_1 = x_u$, $y_1 = y_u$, $x_2 = x_g$, $y_2 = y_g$, $x_3 = x_c$, $y_3 = y_c$ and $x_4 = x_s$, $y_1 = y_s$. Then, CH runs Shamir's $(4,4)$-*threshold* secret sharing scheme for splitting S' as described in Section II and computes new shares $(v_u^\star, v_g^\star, v_c^\star, v_s^\star)$ of the secret S' and holds $v_c^\star$. As a result, computing the new shares for entities make our proposal scheme proactive. Now, CH computes the session key SK_c and sends it through Msg_5 to SN.

Algorithm 5 Step 5: $CH \rightarrow GW$ & SN

```
Input: $p$, $M_c$, $Msg_4$
Output: $Msg_5$
 1: Generates Time-stamp $T_5$
 2: if $|T_4 - T_5| > \Delta T$ then return $\perp$
 3: else
 4:     $v'_s \leftarrow M_{13} \oplus h(M'_s \| M_c \| T_4)$, $k'_s \leftarrow M_{14} \oplus h(M_c \| T_4)$
 5:     $M'_{15} \leftarrow h(M_{13} \| M_c \| T_4 \| k'_s)$
 6:     if $M'_{13} \neq M_{13}$ then return $\perp$
 7:     else
 8:         SN is authenticated
 9:         Computes $S' = f(0) = \sum_{i=1}^{4} y_i \prod_{k=1, k\neq i}^{4} \frac{x_k}{x_k - x_i} \ mod \ p$
10:         $SK_c \leftarrow h(k'_u \| k'_s \| S \| M'_s)$
11:         Computes new shares $(v_u^\star, v_g^\star, v_c^\star, v_s^\star)$ of the secret $S'$
12:         $M_{16} \leftarrow v_s^\star \oplus h(M_c \| T_5 \| M'_s \| k'_s)$
13:         $M_{17} \leftarrow SK_c \oplus h(M'_s \| T_5 \| M_c \| k'_s)$
14:         $M_{18} \leftarrow h(M_{16} \| SK_c \| T_5 \| M'_s \| k'_u \| v_s^\star \| M_{17})$
15:         return $Msg_5 = M_{16}, M_{17}, M_{18}, T_5$
16:     end if
17: end if
```

Algorithm 6 Step 6: SN

```
Input: $M_s$, $M'_c$, $k_s$, $k'_u$, $Msg_5$
Output: $SK_s$, $v_s^\star$
 1: Generates Time-stamp $T_6$
 2: if $|T_5 - T_6| > \Delta T$ then return $\perp$
 3: else
 4:     $v_s^\star \leftarrow M_{16} \oplus h(M'_c \| T_5 \| M_s \| k_s)$
 5:     $SK_s \leftarrow M_{17} \oplus h(M_s \| T_5 \| M'_c \| k_s)$
 6:     $M'_{18} \leftarrow h(M_{16} \| SK_s \| T_5 \| M_s \| k'_u \| v_s^\star \| M_{17})$
 7:     if $M'_{18} \neq M_{18}$ then return $\perp$
 8:     else
 9:         $M_{19} \leftarrow h(T_6 \| v_s^\star \| T_5 \| SK_s \| k_s)$
10:         Share is updated and session Key is accepted
11:         return $SK_s$, $v_s^\star$, $Msg_6 = M_{19}, T_6$
12:     end if
13: end if
```

Step 6 After receiving the message Msg_5, SN runs Algorithm 6 and if both T_5 and M_{18} are valid, SN updates the current share v_s by the new share $v_s^\star$ and accepts the session key SK_s. Otherwise the protocol is terminated.

Step 7 After receiving the message Msg_6, CH runs Algorithm 7 and verifies values of T_6 and M_{19}. If valid, CH accepts the session key and computes M_{20}, M_{21}, M_{22} and M_{23}, and sends them through Msg_7 to GW. In these messages, CH computes $g^{s'} \ mod \ p$ to prove the correctness of S' computed with the received and owned shares.

Step 8 Once the message Msg_7 is received, the GW runs Algorithm 8. If the Time-stamp condition is fulfilled, GW terminates the connection. Otherwise, it obtains SK_g, $v_g^\star$, and $v_u^\star$ and then checks validity of the received M_{17}. If so, CH is mutually authenticated. At this point, GW updates M_u and f_u as $M_{u_{new}}$ and $f_{u_{new}}$ following the equations presented in Algorithm 8, Line 10. It then sends Msg_7 to the user.

Algorithm 7 Step 7: $CH \rightarrow GW$ & SN

```
Input: $p$, $g$, $k'_s$, $SK_c$, $v_u^\star$, $v_g^\star$, $Msg_6$
Output: $SK_c$, $Msg_7$
 1: Generates Time-stamp $T_7$
 2: if $|T_6 - T_7| > \Delta T$ then return $\perp$
 3: else
 4:     $M'_{19} \leftarrow h(T_6 \| v_s^\star \| T_5 \| SK_c \| k'_s)$
 5:     if $M'_{19} \neq M_{19}$ then return $\perp$
 6:     else
 7:         Session Key is accepted
 8:         $M_{20} \leftarrow SK_c \oplus h(g^{S'} \ mod \ p \| T_7)$
 9:         $M_{21} \leftarrow v_g^\star \oplus M_{20} \| T_7)$, $M_{22} \leftarrow v_u^\star \oplus M_{21} \| T_7)$
10:         $M_{23} \leftarrow h(M_{20} \| M_{21} \| M_{22} \| SK_c \| g^{S'} \ mod \ p \| T_7)$
11:         return $SK_c$, $Msg_7 = M_{20}, M_{21}, M_{22}, M_{23}, T_7$
12:     end if
13: end if
```

Algorithm 8 Step 8: $GW \rightarrow User$

```
Input: $g^S \ mod \ p$, $f'_u$, $k'_u$, $Msg_7$
Output: $Msg_8$
 1: Generates Time-stamp $T_8$
 2: if $|T_7 - T_8| > \Delta T$ then return $\perp$
 3: else
 4:     $SK_g \leftarrow M_{20} \oplus h(p^S \ mod \ n \| T_7)$
 5:     $v_g^\star \leftarrow M_{21} \oplus M_{20} \| T_7)$, $v_u^\star \leftarrow M_{22} \oplus M_{21} \| T_7)$
 6:     $M'_{23} \leftarrow h(M_{20} \| M_{21} \| M_{22} \| SK_g \| g^{S'} \ mod \ p \| T_7)$
 7:     if $M'_{23} \neq M_{23}$ then return $\perp$
 8:     else
 9:         CH is authenticated
10:         $M_{u_{new}} \leftarrow h(f'_u \| k'_u)$, $f_{u_{new}} \leftarrow h(M_{u_{new}} \| s)$
11:         $M_{24} \leftarrow M_{u_{new}} \oplus h(f'_u \| T_8)$
12:         $M_{25} \leftarrow f_{u_{new}} \oplus h(M_{u_{new}} \| k'_u \| T_8)$
13:         $M_{26} \leftarrow SK_g \oplus h(f_{u_{new}} \| T_8 \| M_{u_{new}})$
14:         $M_{27} \leftarrow v_u^\star \oplus h(M_{u_{new}} \| T_8 \| k_{u_{new}})$
15:         $M_{28} \leftarrow h(v_u^\star \| M_{u_{new}} \| SK_g \| f_{u_{new}} \| T_6)$
16:         return $Msg_8 = M_{24}, M_{25}, M_{26}, M_{27}, M_{28}, T_8$
17:     end if
18: end if
```

Algorithm 9 Step 9: $User$

```
Input: $f_u$, $M_u$, $Msg_8$
Output: $SK_u$, $v_u^\star$
 1: Generates Time-stamp $T_9$
 2: if $|T_8 - T_9| > \Delta T$ then return $\perp$
 3: else
 4:     $M_{u_{new}} \leftarrow M_{24} \oplus h(f_u \| T_8)$, $f_{u_{new}} \leftarrow M_{25} \oplus h(M_{u_{new}} \| k_u \| T_8)$
 5:     $SK_u \leftarrow M_{26} \oplus h(f_{u_{new}} \| T_8 \| M_{u_{new}})$
 6:     $v_u^\star \leftarrow M_{27} \oplus h(M_{u_{new}} \| T_8 \| k_{u_{new}})$
 7:     $M'_{28} \leftarrow h(v_u^\star \| M_{u_{new}} \| SK_u \| f_{u_{new}} \| T_8)$
 8:     if $M'_{28} \neq M_{28}$ then return $\perp$
 9:     else
10:         $GW$ is authenticated, $SK_u$ is accepted and the share is updated
11:         return $SK_u$, $v_u^\star$
12:     end if
13: end if
```

Step 9 Finally, upon receiving Msg_8, the user executes Algorithm 9 and checks the validity of T_8 and M_{28}. If yes, the user authenticates GW,accepts the session key SK_u, and updates the share. Now, all the entities accept $SK_u = SK_g = SK_c = SK_s$ as a session key and also update their shares.

IV. PAKIT Evaluation

In this section, we aim to evaluate the PAKIT scheme in both security and performance points of view. Regarding security evaluation, we formally/informally analyze the PAKIT scheme. In this paper, we employ the ProVerif [13] language to verify our proposed protocol formally.

A. Informal security analysis

In this section, we informally analyze the security of PAKIT against important and well-known attacks in IoT systems. We also explore the robustness of PAKIT against an adversary who compromised a cluster head/ a sensor node/a gateway node.

1) Man-in-the-middle Attack: In our proposed protocol, if each entity cannot validate the integrity message (e.g., M_4, M_9, f_s, M_{13}, M_{18}, M_{19}, M_{20} and M_{28}), it terminates the connection and does not pass the authentication step. Nevertheless, an adversary who does not have any knowledge on f_u, GW's private key (s), mask the identity of CH, mask the identity of SN, and current session key, cannot compute these messages. So, an adversary cannot forge any valid entity to cheat other entities and authenticate her/him to them. Therefore, PAKIT scheme can provide mutual authentication, and no entity can cheat another entity and execute man-in-the-middle attack.

2) Impersonation Attack: On this occasion the adversary aims to cheat GW, CH and SN. S/he may use the eavesdropped first message $\langle Msg_1 = M_u$, M_1, M_2, M_3, M_4, $T_1\rangle$ of the previous sessions to impersonate a GW. Once the eavesdropped message is received, the GW checks the legitimacy of the user U_i by validating T_1 and $M_4 = h(M_u\|k_u\|f_u\|T_1\|M_s\|M_3)$. The adversary has to posses f_u and fresh T_1 to forge M_4. Even if s/he can generate acceptable T_1, without having any knowledge about the GW's master key (s), the adversary A cannot calculate a valid M_4. Moreover, to impersonate a gateway, the adversary has to forge the messages $\langle Msg_2 = M_5$, M_6, M_7, M_8, M_9, M_{10}, $T_2\rangle$ or $\langle Msg_8 = M_{24}$, M_{25}, M_{26}, M_{27}, M_{28}, $T_8\rangle$. So, the adversary A needs to know f_c, SK_g and $f_{u_{new}}$ to compute $M_9 = h(v_g\|f_c\|M_8\|T_2\|v_u)$ and $M_{28} = h(v_u^{\star}\|M_{u_{new}}\|SK_g\|f_{u_{new}}\|T_6)$ which is impossible. Thus, A cannot forge the aforementioned messages. Besides, if an adversary wants to impersonate a cluster head, s/he tries to pass the SN and GW verification phase. To achieve this malicious goal, the adversary has to forge the messages $\langle Msg_3 = M_{10}, M_{11}, M_{12}, T_2, T_3\rangle$ or $\langle Msg_5 = M_{16}, M_{17}, M_{18}, T_5\rangle$ for cheating SN or $Msg_7 = M_{20}, M_{21}, M_{22}, M_{23}, T_7$ for cheating GW. So, the adversary A needs to know M_c, M_s and S which is impossible. Thus, A cannot forge the aforementioned messages. Finally, in the PAKIT scheme, SN computes $\langle Msg_4 = M_{13}, M_{14}, M_{15}, T_4\rangle$ and $\langle Msg_6 = M_{19}, T_6\rangle$ and sends these to the CH in Phase 4 and Phase 5, respectively. To forge the messages $M_{13} = v_s \oplus h(M_s\|M_c\|T_4)$ and $M_{19} = h(T_6\|v_s^{\star}\|T_5\|SK_s\|k_s)$, the adversary A must know M_c, M_s and the current session key. Moreover, A cannot compute the secret shares v_s and $v_s^{\star}$. Therefore, A cannot compute SN's messages to execute a sensor node impersonation attack. Therefore, PAKIT scheme can resist against GW, CH, and SN impersonation attacks.

3) Session Key Security: In the PAKIT scheme, the attacker can eavesdrop all the messages that are included in the session key. Nevertheless, the session key $SK_s = SK_c = SK_g = SK_u$ is supported by the one-way hash function $h(\cdot)$. Thus, it is computationally impossible for the attacker to obtain the session key. Besides, thanks to the fresh random numbers employed in computing the session key (e.g., k_u and k_s) and the one-way hash function; even if an adversary has the previous session key s/he cannot compute current session key (forward secrecy) and also if s/he has the current session key, s/he cannot computes next protocol's session key (backward secrecy). Thus, PAKIT scheme has session key security.

4) Anonymity: In PAKIT scheme, the identity of each entity, for example, ID_u is never sent in plain-text over an insecure channel. In this sense, the masked identity, for example, $M_u = h(ID_u\|r_u)$ of the user is the value passed through the insecure channel. Due to the collision-resistant property of the one-way hash function $h(\cdot)$, obtaining the identity from masked identity is computationally impossible for the adversary. Therefore, PAKIT scheme is successful to preserve the anonymity of the entities.

5) Preserving user untraceability: In this attack, an adversary A tries to determine whether the same (unknown) user computes two messages. Fortunately, in PAKIT scheme all the parameters used in the messages $\langle Msg_1 = M_u, M_1, M_2, M_3, M_4, T_1\rangle$ are randomly generated. Moreover, when the update phase of the PAKIT scheme is executed, the user updates the value of M_u for the future. Therefore, A cannot decide whether two different sessions of the protocol are associated with the same user. Thus, in our PAKIT scheme, A cannot track the user.

6) Replay attack: If an adversary can deceive legitimate entities by forwarding the eavesdropped messages of the previous sessions of the protocol, the protocol suffers from a replay attack. In PAKIT protocol the time-stamps and random numbers used in all messages of the protocol make it robust against a replay attack.

7) Entity Compromised Attack: In our proposal, if an adversary compromises an entity involved in the scheme (i.e., SN, CH and GW) and obtains all stored secrets, s/he cannot compute $S = f(0) = \sum_{i=1}^{4} y_i \prod_{k=1,k\neq i}^{4} \frac{x_k}{x_k - x_i}\ mod\ p$ (note that $x_1 = x_u$, $y_1 = y_u$, $x_2 = x_g$, $y_2 = y_g$, $x_3 = x_c$, $y_3 = y_c$ and $x_4 = x_s$, $y_1 = y_s$) and the session key without having any information about these parameters. In addition, these parameters are updated in each session thanks to the proactive property of the Shamir's secret sharing scheme. Furthermore, in the GW's side, the value of the secret S is never stored, but the value of $g^S\ mod\ p$ is stored in the memory of the GW. So, if an adversary can access to this value by compromising GW, because of the (DL) problem presented in Section II, s/he cannot obtain S. Thus, the proposed protocol is secure against the entity compromised Attack.

B. Formal security analysis

For formal security analysis, we adopt ProVerif language [13]. In the ProVerif tool, we implement PAKIT authentication protocol. We present the obtained results in Fig 2. In Fig 2, the first four results demonstrate shared session keys SK_u, SK_g, SK_c and SK_s. The adversary is unable to access these keys. Besides, the rest of the results present the results of two injective correspondence assertions which can prove the mutual authentication of the PAKIT.

```
Query not attacker(SKu[])
RESULT not attacker(SKu[]) is true.
Query not attacker(SKg[])
RESULT not attacker(SKg[]) is true.
Query not attacker(SKc[])
RESULT not attacker(SKc[]) is true.
Query not attacker(SKs[])
RESULT not attacker(SKs[]) is true.
Query inj-event(UserAuth(id)) ==> inj-event(UserLogin(id))
RESULT inj-event(UserAuth(id)) ==> inj-event(UserLogin(id)) is true.
```

Fig. 2: The achieved results from the ProVerif language.

We compare the security features offered by our proposed protocol and other similar ones in Table II, where the symbol "Y" indicates that the scheme is secure against the related attack and the symbol "N" indicates the contrary. Also, the symbol "ND" indicates that the pointed feature is not defined for the related scheme. We can conclude that our PAKIT satisfies all the security features required and offers a higher security level than its predecessors.

TABLE II: Security features comparison.

Security features	[6]	[7]	[8]	[9]	PAKIT
Protection of user untraceability	Y	N	Y	Y	**Y**
Resistance against replay attack	Y	N	Y	Y	**Y**
Resistance against impersonation attacks	N	N	Y	Y	**Y**
Resistance against entity compromised attack	N	N	ND	ND	**Y**
Anonymity preserving	Y	N	ND	ND	**Y**
Resistance against identity guessing attack	Y	N	N	Y	**Y**
Providing session key security	Y	Y	Y	Y	**Y**

C. Performance Analysis

Due to the limitation of the resource-constraint sensors employed in the IoT systems, the performance compression between our proposed scheme with the other state-of-the-art schemes is presented in terms of the sensor side.

TABLE III: Sensor node computational overhead.

Scheme	Sensor node	Execution Time (ms)
He *et al.* [6]	$T_h + 2T_s$	0.261
Yeh *et al.* [7]	$2T_{Exp} + 3T_h$	23.381
Kumar *et al.* [8]	$T_h + 2T_s$	0.261
Wazid *et al.* [9]	$9T_h$	0.0036
PAKIT	$\mathbf{10T_h}$	**0.004**

in ECC algorithm execution time, and $T_s = 0.1303$ ms is

Table III presents the performance comparison of our proposed PAKIT scheme against the most relevant state-of-the-art IoT lightweight authentication protocols such as He *et al.* [6], Yeh *et al.* [7], Kumar *et al.* [8], Wazid *et al.* and [9]. In this table, $T_h = 0.0004$ ms is 160 bits SHA-1 execution time, $T_{Exp} = 11.69$ ms is modular exponential operation AES encryption/decryption execution time [15]. In PAKIT, the sensor node executes algorithms 4 and 6 in which the number of hash values that the sensor node should compute are six and four, respectively. From Table III we conclude that our proposal is still efficient enough.

V. Conclusion and Future Work

In this paper, we propose a novel authentication and key agreement scheme called PAKIT in IoT system. Our protocol is secure against entity compromised attack, and can effectively provide secure communications in IoT systems. We can use PAKIT for specific fields like health care, e-commerce, energy, vehicle access. Furthermore, in PAKIT we use secret sharing to remedy the compromised entity challenge. We update all the shares without changing the secret (proactive secret sharing) to ensure that old compromised shares are not valid to construct secret content. Additionally, we informally and formally(using ProVerif language) evaluate the security of PAKIT scheme and show that it can properly withstand the known attacks and it is efficient enough in the sensor node side. In the future, we focus on designing a secure lightweight group-entity authentication and key agreement protocol in which several numbers of cluster heads and sensor nodes are considered.

References

[1] M. A. Ferrag *et al.*, "Authentication protocols for internet of things: a comprehensive survey," *Security and Communication Networks*, vol. 2017, 2017.

[2] P. Syverson, "A taxonomy of replay attacks [cryptographic protocols]," in *IEEE CSFW'7*, 1994, pp. 187–191.

[3] E. E. Schultz, "A framework for understanding and predicting insider attacks," *Computers & Security*, vol. 21, no. 6, pp. 526–531, 2002.

[4] W.-C. Ku *et al.*, "Impersonation attack on a dynamic id-based remote user authentication scheme using smart cards," *IEICE Transactions on Communications*, vol. 88, no. 5, pp. 2165–2167, 2005.

[5] J. Jung *et al.*, "An anonymous user authentication and key agreement scheme based on a symmetric cryptosystem in wireless sensor networks," *Sensors*, vol. 16, no. 8, p. 1299, 2016.

[6] D. He *et al.*, "Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks," *Multimedia Systems*, vol. 21, no. 1, pp. 49–60, 2015.

[7] H.-L. Yeh *et al.*, "A secured authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 11, no. 5, pp. 4767–4779, 2011.

[8] P. Kumar *et al.*, "E-sap: efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks," *Sensors*, vol. 12, no. 2, pp. 1625–1647, 2012.

[9] M. Wazid *et al.*, "Design of secure key management and user authentication scheme for fog computing services," *Future Generation Computer Systems*, vol. 91, pp. 475–492, 2019.

[10] L. Wu *et al.*, "Secure key agreement and key protection for mobile device user authentication," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 319–330, 2019.

[11] Y. Liu *et al.*, "Double verification protocol via secret sharing for low-cost RFID tags," *Future Generation Computer Systems*, vol. 90, pp. 118–128, 2019.

[12] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[13] B. Blanchet, "Automatic verification of security protocols in the symbolic model: The verifier proverif," in *Foundations of Security Analysis and Design VII*. Springer, 2014, pp. 54–87.

[14] D. Dolev *et al.*, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[15] L. Xu *et al.*, "Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care," *Journal of medical systems*, vol. 39, no. 2, p. 10, 2015.