Journal of Theoretical and Applied Information Technology
15th February 2018. Vol.96. No 3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

LIGHTWEIGHT IDENTITY BASED SIGNATURE FOR MOBILE OBJECT AUTHENTICATION IN THE INTERNET OF THINGS

[1] MAHA SAADEH, [2] AZZAM SLEIT, [3]KHAIR EDDIN SABRI, [4]WESAM ALMOBAIDEEN

[1]Ph.D candidate. The University of Jordan, Department of Computer Science, Jordan
[2]Professor. The University of Jordan, Department of Computer Science, Jordan
[3]Associate Professor. The University of Jordan, Department of Computer Science, Jordan
[4]Professor. The University of Jordan, Department of Computer Science, Jordan
E-mail: [1]mha9130287@fgs.ju.edu.jo, [2]azzam.sleit@ju.edu.jo, [3]k.sabri@ju.edu.jo,
[4]almobaideen@inf.ju.edu.jo

ABSTRACT

Trusted communication is crucial for data sharing and resource access in the context of the Internet of Things (IoT). This paper presents a lightweight hierarchical authentication protocol, using identity based signature, to serve IoT mobile objects. The proposed protocol has three entities; Private Key Generator (PKG), sub_PKG, and mobile objects. A comparison with other related protocols according to the key generation method, key distribution method, and the security attack model is presented. BAN logic is used for formal verification of the proposed protocol. Moreover, the performance is evaluated based on a quantitative measure of performance metrics such as number of scalar multiplication and modular inverse operations. The evaluation shows that the proposed protocol has a lower total computation cost since it does not use expensive hash to point, modular inverse, and bilinear pairing operations. This makes it more efficient and suitable in supporting IoT constrained mobile objects.

Keywords: *Internet of Things, Hierarchical Architecture, Object Authentication, Identity Based Signature, Object Mobility.*

1. INTRODUCTION

Connecting extremely large number of stationary and mobile objects of various types, sizes, and functionalities will form the biggest network known as the Internet of Things (IoT). These objects are connected in order to offer different services such as data access and sharing among IoT objects. IoT services are facilitated by the development of smart technologies such as smartphones, smart homes, and smart communities [1] [2] [3] [4]. On the other hand, offering these services has raised crucial security issues. One main security issue is the trustworthiness between connected objects in order to allow data sharing. This trust can be facilitated by providing authentication services among IoT objects [4].

In this paper, a lightweight authentication protocol based on Elliptic Curve Cryptography (ECC) is proposed. The main objective is to support authentication of constrained mobile objects in the context of IoT. Since these devices are limited in their storage and computation capabilities, the proposed authentication protocol provides a lightweight signature solution. The authentication method is performed using Hierarchical Identity Based Signature (HIBS) [5]. It consists of five phases; the PKG setup phase, the sub_PKG setup phase, the extract phase, the signing phase, and the verification phase.

Identity Based Signature (IBS) was first proposed by Shamir [6] in which user's identity is used to generate the signing and verification keys. The entity which is responsible for keys generation is called private key generator [7] [8]. IBS has the advantage of eliminating certificate management since, unlike traditional public key digital signature protocols, certificates are not required [7]. This reduces the overhead on IoT constrained devices by eliminating the computational overhead of certificate validation and the storage space that is needed to store object's certificates.

Identity based protocols that have been proposed in the literature are mainly based on bilinear paring or ECC. Bilinear pairing based protocols are considered to be computationally

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

expensive [9]. Consequently, the authentication protocol that is proposed in this paper is based on ECC in order to support IoT constrained devices.

The proposed protocol is compared with other ECC and bilinear based protocols according to several factors which are key generation methods, key distribution methods, and the security attack model. Formal verification of the proposed protocol is done using BAN logic [10]. Moreover, the performance is evaluated based on a quantitative measure of computation cost metrics such as the number of scalar multiplication and modular inverse operations. Results show that the proposed protocol has the lowest cost among the compared protocols since it does not use expensive computation operations such as hash to point operation, modular inverse operation, and bilinear pairing operation [11] [12]. This makes it more efficient and suitable for IoT constrained devices.

The rest of the paper is organized as follows: Section 2 summarizes the related works. The details of the proposed authentication protocol are discussed in Section 3. Section 4 discusses the security analysis for the proposed protocol. Comparison and performance evaluation are discussed in Section 5. Finally, Section 6 concludes the paper.

2. RELATED WORKS

Several identity based protocols have been proposed in the literature. In this section some related hierarchical protocols are summarized.

In [13], the authors proposes elliptic curve digital signature algorithm for vehicular networks. It has three entities; the central authority, the Road Side Unit (RSU) and the On Board Unit (OBU). System's public parameters and all RSU public keys are generated by the central authority in the setup phase. These keys are transmitted securely to each RSU. The OBU uses the RSU public key in order to verify the RSU generated signatures. Although this protocol is based on ECC, multiple complex modular inverse operations are required to perform the authentication process. Another signature protocol vehicular networks is proposed in [14]. It is based on IBS and uses hash chains and bilinear pairing in signature keys generation. This protocol reduces the overload on the trusted authority by allowing vehicles to update their credentials on road through RSUs. This protocol has the same disadvantages since it does not support constrained devices and it is based on bilinear pairing which is considered to be expensive operation compared with ECC.

The protocol proposed in [15] has three layers and is proposed for automatic dependent surveillance-broadcast systems. A PKG, which generates and publishes system public parameters, is located in the first layer. The second and third layers consist of airlines and aircrafts, respectively. Airlines and aircrafts should be registered in the hierarchy in order to get their private keys. For aircrafts authentication, the signer aircraft issues a digital signature using its private key and the verifier aircrafts verifies the signature based on the signer aircraft's ID and the public parameters. This protocol has the same disadvantages of [14] discussed above.

The IBS protocols in [5] and [16] have been proposed for cloud computing systems. The protocol in [5] has three levels. A root PKG, which allocates identities for registered clouds, is located at Level-0. Level-1 contains the cloud systems. End users and servers are located at Level-2. Cloud systems are responsible for managing identities for end users and servers. In [16] the root PKG is called the broker and sub-PKGs are Cloud Security Administrators (CSA). Users and clouds are registered to CSAs and their information will be stored at the broker repository. For authentication the user enters his identity. Afterwards, the authority proof is requested from the broker through the parent CSA. After that, the broker sends the user authority proof along with a generated private key. These protocols do not support constrained devices, do not support mobile objects, and they are based on expensive bilinear pairing.

In summary, the discussed protocols have complex computational operations that introduce considerable overhead on IoT constrained devices. In this paper, a lightweight ECC based authentication protocol for mobile object that is connected with the IoT is proposed. In the next section, the details of the proposed protocol are discussed.

3. THE PROPOSED IDENTITY BASED SIGNATURE PROTOCOL

The proposed identity based signature protocol is based on ECC and consists of five main phases. The first phase is the PKG setup phase in which the PKG selects the elliptic curve public parameters and its own private and public keys. The sub_PKG setup is

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the second phase in which each sub_PKG contacts the PKG in order to obtain its private key and the public parameters. During the third phase, which is known as the extract phase, the mobile object contacts with one sub_PKG in order to obtain its private key and the public parameters. In the signing phase, the fourth one, the mobile object uses his private key to generate a signature denoted as **σ**. Finally, in the verification phase, the verifier uses the public key of the mobile object in order to verify the signature **σ**.

3.1 Preliminaries

3.1.1 Elliptic Curve Cryptography

An Elliptic Curve *G* over a finite field *Fq* is defined by $y^2 = x^3 + ax + b$, where $4a^3 + 27b^2 \not\equiv 0 (mod\ q)$ and *a, b* are two coefficients that define the curve equation such that *a, b* are integers **ϵ** *Fq* [17]. *q* is a prime number such that $q > 3$. The curve cofactor $h = \#E(Fq)/n$. Where $\#E(Fq)$ is the number of curve points over *Fq*. Each point on the elliptic curve is an element of *Fq* represented by (*x, y*) where both *x* and *y* **ϵ** *Fq*.

The following are the set of operations that are defined over elliptic curves:

1- Point addition: let R and S be two points on G such that R ≠ S. Then, Q = R + S is calculated by Equation (1).

$$\begin{pmatrix} R_x \\ R_y \end{pmatrix} + \begin{pmatrix} S_x \\ S_y \end{pmatrix} = \begin{cases} s = \frac{R_y - S_y}{R_x - S_x} \\ Q_x = s^2 - R_x - S_x \\ Q_y = s.(R_x - Q_x) - R_y \end{cases} \quad (1)$$

2- Point doubling: let R be a point on G. Then, Q = R + R is calculated by Equation (2).

$$\begin{pmatrix} R_x \\ R_y \end{pmatrix} + \begin{pmatrix} R_x \\ R_y \end{pmatrix} = \begin{cases} s = \frac{3 R_x^2 + a}{2 R_y} \\ Q_x = s^2 - 2R_x \\ Q_y = s.(R_x - Q_x) - R_y \end{cases} \quad (2)$$

3- Scalar multiplication: let R be a point on G. Then, Q = k R is calculated by Equation (3) where k is an integer. Scalar multiplication is calculated as a sequence of point doubling. Scalar multiplication is important for the security of ECC based signature protocols. For example if Q = k R, then it is hard to find k from both Q and R. So k is secret. This problem is called the Elliptic Curve Discrete Logarithm Problem (ECDLP). For more details on elliptic curves refer to [17].

$$k R = \overbrace{R + R + R + \cdots + R}^{\text{k times}} \quad (3)$$

3.1.2 BAN Logic

BAN logic is a logic proposed by Burrows, Abadi, and Needham [10] to analyze the security of authentication protocols. It is based on protocol's initial assumptions and uses inference rules to infer other facts in order to achieve authentication goals.

The following are the main notations and inference rules in BAN logic [10], [18], and [19]:

Notations:

1. P|≡ X: P believes X. that means that P considers X to be true and acts based on this.
2. P ◁ X: P sees X, for example when P received a message contains X, then P sees X.
3. P |~ X: P said X.
4. P ||~ X: P recently said X.
5. P ⇒ X: P has jurisdiction over X or P controls X.
6. #(X): The formula X is fresh. That is X has not been sent in a message at any time before the current run of the protocol.
7. $P \overset{k}{\leftrightarrow} Q$: k is a symmetric key that is used by P and Q to communicate.
8. $\overset{k}{\rightarrow}P$: k is P's public key.
9. $P \overset{X}{\rightleftharpoons} Q$: The formula X is a secret known only to both P and Q.
10. $\{x\}_k$: This represents that formula X is encrypted using the key k.
11. $\{x\}_k^{-1}$: This represents that formula X is encrypted using the inverse key of k. i.e. if k is a public key, then k^{-1} is the private key corresponds to k.
12. $<x>_Y$: This represents X combined with formula Y; it is intended that Y to be a secret and the presence of Y authenticates who ever utters $<x>_Y$.
13. PK (k, P): k is a public key for P and there exists a unique key corresponds to k.
14. II(P): P has a private key which is known only to P.
15. **σ**(X, P): X is signed by P's private key.

Inference Rules:

Symmetric Rules: (R1) [10] if P believes that k is a good symmetric key which is shared with Q, and P sees a formula x that is encrypted with k, then P believes that Q said x. (R2) [10] if P believes that y is a shared secret with Q, and P sees a formula x that is combined with y, then P believes that Q said x.

$$(R1): \frac{P \mid\equiv P \overset{k}{\leftrightarrow} Q, P \lhd \{x\}_k}{P \mid\equiv Q \mid\sim x}$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$(R2): \frac{P \mid\equiv P \overset{y}{\rightleftharpoons} Q, P \lhd \langle x \rangle_y}{P \mid\equiv Q \mid\sim x}$$

Asymmetric Rule [10]: if P believes that k is a good public key for Q, and P sees a formula x that is encrypted with k^{-1}, then P believes that Q said x.

$$(R3): \frac{P \mid\equiv \overset{k}{\rightarrow} Q, P \lhd \{x\}_{k^{-1}}}{P \mid\equiv Q \mid\sim x}$$

Freshness Rules: (R4) [10] if P believes that formula x is fresh, and P also believes that Q said x, then P believes that Q believes x. (R5) [18] if P believes that formula x is fresh, and P also believes that Q said x, then P believes that Q has recently said x. (R6) [18] if P believes that formula x is fresh, then P believes that any formula combined with x is fresh.

$$(R4): \frac{P \mid\equiv \#(x), P \mid\equiv Q \mid\sim x}{P \mid\equiv Q \mid\equiv x}$$

$$(R5): \frac{P \mid\equiv \#(x), P \mid\equiv Q \mid\sim x}{P \mid\equiv Q \mid\mid\sim x}$$

$$(R6): \frac{P \mid\equiv \#(x)}{P \mid\equiv \#(x,y)}$$

Synthetic Rule [18]: if P believes that part of formula x is fresh, then P believes that formula x is fresh.

$$(R7): \frac{P \mid\equiv \#(x')}{P \mid\equiv \#(x)}$$

Seeing Rules: (R8) [18] if P sees formula (x, y), then P sees formula x and also P sees formula y. (R9) [10] if P sees formula x combined with secret y that P has, then P sees formula x. (R10) [10] if P believes that k is a good symmetric key which is shared with Q, and P sees a formula x that is encrypted with k, then P sees x. (R11) [10] if P believes that k is a good public key for P, and P sees a formula x that is encrypted with k, then P sees x. (R12) [10] if P believes that k is a good public key for Q, and P sees a formula x that is encrypted with k^{-1}, then P sees x.

$$(R8): \frac{P \lhd (X,Y)}{P \lhd (X),\ P \lhd (Y)}$$

$$(R9): \frac{P \lhd \langle x \rangle_y}{P \lhd x}$$

$$(R10): \frac{P \mid\equiv P \overset{k}{\leftrightarrow} Q, P \lhd \{x\}_k}{P \lhd x}$$

$$(R11): \frac{P \mid\equiv \overset{k}{\rightarrow} P, P \lhd \{x\}_k}{P \lhd x}$$

$$(R12): \frac{P \mid\equiv \overset{k}{\rightarrow} Q, P \lhd \{x\}_{k^{-1}}}{P \lhd x}$$

Jurisdiction Rule [10]: if P believes that Q controls x, and P believes that Q believes x, then P believes x.

$$(R13): \frac{P \mid\equiv Q \Rightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$$

Signing Rules: (R14) [19] if P believes that k is a public key for Q, and P also believes that Q has a private key corresponds to k, and P sees formula x signed by Q's private key, then P believes that Q said x. (R15) [19]: if P sees formula x signed by Q's private key, the P sees x.

$$(R14): \frac{P \mid\equiv PK(k,Q), P \mid\equiv II(Q), P \lhd \sigma(X,Q)}{P \mid\equiv Q \mid\sim X}$$

$$(R15): \frac{P \lhd \sigma(X,Q)}{P \lhd X}$$

Other Rules: (R16) [20] if P believes x, and P also believes y, then P believes the formula (x, y). (R17) [20] if P believes formula (x, y), then P believes x. (R18) [20] if P believes that Q believes the formula (x, y), then P believes that Q believes x. (R19) [18]if P believes that Q said formula (x, y), then P believes that Q said x.

$$(R16): \frac{P \mid\equiv x, P \mid\equiv y}{P \mid\equiv (x,y)}$$

$$(R17): \frac{P \mid\equiv (x,y)}{P \mid\equiv x}$$

$$(R18): \frac{P \mid\equiv Q \mid\equiv (x,y)}{P \mid\equiv Q \mid\equiv x}$$

$$(R19): \frac{P \mid\equiv Q \mid\sim (x,y)}{P \mid\equiv Q \mid\sim x}$$

For more information about BAN logic, refer to [10].

3.2 The Main Phases of the Proposed Protocol

The propose protocol is based on a hierarchical architecture in order to increase its scalability by distributing objects and systems over several sub-PKG. This reduces the overhead on the PKG. The assumed hierarchy consists of a PKG in the top first level, set of sub-PKGs in the middle second level, and objects and systems in the third bottom level. The PKG manages the whole domain and, accordingly, each sub-PKG in the domain should be associated with the PKG. In order to be authenticated, all mobile objects should register their identity to one of the sub-PKG. The identity of the mobile object is mapped to the identities of its ancestors in the hierarchy. For example, if the PKG identity is IDr, then sub-PKG *s* identity is (IDr || IDs) and object *o* identity is (IDr || IDs|| IDo). Figure 1 shows the architecture of the proposed protocol.

Journal of Theoretical and Applied Information Technology
15th February 2018. Vol.96. No 3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

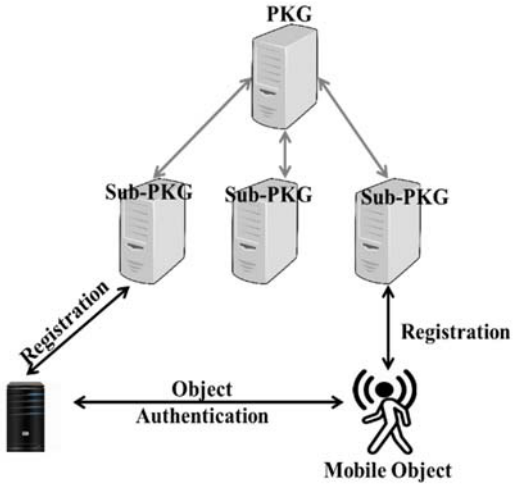

Figure 1: The Architecture of the Proposed Protocol.

Following are the details of the five phases that the proposed protocol consists of.

Table 1: List of Symbols.

Symbol	Description
G	An elliptic curve Group
q	The size of the finite field, Large Prime, and $q \neq nh$ (to avoid weak curves)
F_q	Finite field over q
n	The order of the curve generator P, where $n > 2^{160}$
P	The elliptic curve base point of order n, "group generator"
x	PKG master secret key, $x \in [1, n-1]$
Q_0	PKG Public key
H_1	H_1: $\{0,1\}^* \times G \rightarrow Z_n$, a hash function that takes identity in binary representation and a point on the curve and convert it into integer in the interval [1, n-1].
H_2	H_2: $\{0,1\}^* \rightarrow Z_n$, a hash function that takes a nonce value in binary string and convert it into integer in the interval [1, n-1].
a	Field element in F_q to define the curve equation over F_q
b	Field element in F_q to define the curve equation over F_q
k	A nonce generated by the mobile object
h	Cofactor $h = \#E(F_q)/n$
PK	The mobile object private key
Q_p	The signature verification key
Q_{p^*}	The object public key
SK	Sub-PKG private key
Q_s	Sub-PKG public key
M	Message
n_i	A nonce generated by the verifier

3.2.1 PKG setup phase

In this phase the PKG selects the following parameters:

1. Elliptic curve G over a finite field F_q.
2. Two integers a and b that define the elliptic curve over F_q by the equation $y^2 = x^3+ax+b \bmod q$. The two integers $a, b \in F_q$, and $4a^3 + 27b^2 \neq 0 \pmod q$.
3. P as a base point of group G of order n.
4. Random integer $x \in [1, n-1]$.
5. Point $Q_0 \in G$ such that $Q_0 = x P$.
6. Two hash functions; H_1:$\{0,1\}^* \times G \rightarrow Z_n$, and H_2: $\{0, 1\}^* \rightarrow Z_n$. these hash functions are one-way collision resistant functions.
7. Curve cofactor h as $h = \#E(F_q)/n$. where $\#E(F_q)$ is a number of points in G.

Then, the PKG sets $G = \{a, b, P, q, n, h\}$, H_1, H_2, and Q_0 as public parameters P_{pub} and keeps x as its master secret key.

3.2.2 Sub-PKG setup phase

In this phase each sub-PKG registers its identity SID to the PKG, then the PKG does the following.

1. Selects a random integer r such that $r \in [1, n-1]$.
2. Calculates $U_s = r P$
3. Calculates $h_s = H_1(SID \,||\, U_s)$.
4. Calculates Q_s as

$$Q_s = U_s + h_s Q_0 \quad (4)$$

5. Calculates sub-PKG$_i$'s private key as

$$SK = (x h_s + r) \bmod n \quad (5)$$

6. Sends SK and Q_s to sub-PKG through a secure channel along with P_{pub}.

3.2.3 Extract phase

In this phase the mobile object registers its identity ID with the sub-PKG in order to obtain its private key. The mobile object sends its ID to the sub-PKG which generates the object's private key PK as in Equation (6)

$$PK = (SK\, h_p + r_p) \bmod n \quad (6)$$

Where $h_p = H1(ID \,||\, U_p)$, $U_p = r_p P$, and r_p is a random integer such that $r_p \in [1, n-1]$. Then, the sub-PKG calculates the mobile object public key Q_{p^*} as in Equation (7)

$$Q_{p^*} = U_p + h_p Q_s \quad (7)$$

Then, the sub-PKG sends (PK, Q_{p^*}) to the mobile object through a secure channel along with P_{pub}.

3.2.4 Signing phase

In this phase the mobile object signs a message M with signature $\boldsymbol{\sigma}$ as in the following steps:

1. The mobile object sends an access request message to the verifier which represents a contact point to the system with which the mobile object wants to have access.
2. The verifier generates a nonce n_i and calculates $e = H_2(n_i)$. Then, it sends e to the mobile object.
3. The mobile object generates a random nonce k. The generation of k should be unpredictable.
4. The mobile object calculates the verification key Q_p as in Equation (8):
$$Q_p = k \,.\, P \quad (8)$$
5. The mobile object calculates the signature $\boldsymbol{\sigma}$ as in Equation (9) or Equation (10):
$$\boldsymbol{\sigma} = ((e \times PK) + k) \bmod n \quad (9)$$
$$\boldsymbol{\sigma} = (e\,(r_p + h_p\,(r + h_s\,x)) + k) \bmod n \quad (10)$$
6. The mobile object sends ($\boldsymbol{\sigma}$, Q_p, Q_{p^*}) to the verifier.

3.2.5 Verifying phase

In this phase the verifier verifies the signature $\boldsymbol{\sigma}$ based on P_{pub}, Q_p, and Q_{p^*} by verifying that $\sigma . P = e . Q_{p^*} + Q_p$. If yes, then the signature is verified successfully. Otherwise, the verifier rejects the verification. The extract, signing, and verifying phases are illustrated in Figure 2.

4. SECURITY ANALYSIS

In this section, the security of the proposed identity based authentication protocol is analyzed.

4.1 Correctness Proof

The condition $\sigma . P = e . Q_{p^*} + Q_p$ holds when the signature is correct since

$\sigma . P$ $= e . Q_{p^*} + Q_p$,
⟨substitute Q_p from Equation (8)⟩
$= e . Q_{p^*} + k . P$,
⟨substitute Q_p* from Equation (7)⟩
$= e\,(U_p + h_p . Q_s) + k . P$,
⟨substitute Q_s from Equation (4) ⟩
$= e\,(U_p + h_p\,(U_s + h_s . Q_0)) + k . P$,
⟨*substitute* $Q_0 = x . P$, $U_s = r . P$, *and* $U_p = r_p . P$⟩
$= e\,(r_p . P + h_p\,(r . P + h_s\,x . P)) + k . P$,
⟨Take P as common factor⟩
$= e . P\,(r_p + h_p\,(r + h_s\,x)) + k . P$
$= P\,(e\,(r_p + h_p\,(r + h_s\,x)) + k)$,
⟨recall Equation (10) $\sigma = e\,(r_p + h_p\,(r + h_s x)) + k$⟩
$= P . \sigma$

4.2 Attacks Model

Theorem 1: an adversary *A* cannot know the private and secret keys from public information.

Proof: 1. Given Q_0 and P, which are public information, then adversary A cannot find secret key x since it is an ECDLP [21].

2. Given the signature $\boldsymbol{\sigma}$, adversary A cannot calculate the mobile object's private key PK, since $\boldsymbol{\sigma}$ is calculated based on two secret values which are k, and PK that are known only to the mobile object.

3. Knowing two different signatures σ and σ' will not allow adversary A to calculate the mobile object private key PK since each signature is generated based on a new generated nonce k. Consequently, σ and σ' are generated based on two different nonce values; k and k'.

Theorem 2: by knowing the verification key Q_p, the adversary *A* cannot find *k*.

Proof: Q_p is a point on G calculated as $k . P$. Knowing Q_p adversary A cannot find k since it is an ECDLP.

Theorem 3: an adversary *A* cannot link two previously sent signatures σ and σ' in order to calculate mobile object *o* private key *PK*.

Proof: the hashed values e and e' that are used to generate σ and σ', respectively are different since H_2 is a collision resistant hash function. Consequently, signatures σ and σ' are different and adversary A cannot find two equal signatures. This applies to all messages.

Theorem 4: using old valid signatures, an adversary *A* cannot forge the signature of mobile object *o*.

Proof: in order to generate a valid signature, A needs to know k and PK. Both of these information are private. Moreover, according to Theorem 3 and Theorem 1, A cannot link two previous signatures σ and σ' in order to find PK since each time e is calculated based on a new nonce and different k is generated, consequently, $\sigma \neq \sigma'$.

Theorem 5: the proposed IBS protocol is safe against replay attack.

Proof: adversary A cannot replay old signed messages since each signature is generated by

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

a fresh nonce n_i generated by the verifier. If *A* replays an old signed message, the verifier will check the nonce and reject the replayed message.

Theorem 6: adversary *A* cannot know the private key that is used in the signature unless *k* is the same for two different signatures.

Proof: let σ and σ' be two signatures such that $\sigma = ((e \times PK) + k) \bmod n$ and $\sigma' = ((e' \times PK) + k) \bmod n$, from Equation (9), *k* can be calculated as $k = \sigma - (e \times PK) \bmod n$, and $k = \sigma' - (e' \times PK) \bmod n$. Adversary *A* performs the following steps to obtain the private key *PK*:

Step 1: $\sigma - (e \times PK) = \sigma' - (e' \times PK)$
Step 2: $\sigma - \sigma' = (e \times PK) - (e' \times PK)$
Step 3: $\sigma - \sigma' = PK\ (e - e')$
Step 4: $(\sigma - \sigma') / (e - e') = PK$

σ, σ', e, and *e'* are known, so *A* knows the private key of the mobile object. Consequently, *k* must be a nonce value that differs each time it is generated.

Theorem 7: adversary *A* cannot know the private key that is used in the signature unless *k* is predictable.

Proof: let σ be a signatures such that $\sigma = ((e \times PK) + k) \bmod n$. From Equation (9), *k* can be calculated as $k = \sigma - (e \times PK) \bmod n$. Adversary *A* performs the following steps to obtain the private key *PK*:

Step 1: $k = \sigma - (e \times PK)$
Step 2: $k - \sigma = - (e \times PK)$
Step 3: $(\sigma - k) / e = PK$

σ, e, and *k* are known, so *A* knows the private key of the mobile object. Therefore, the generation of *k* must be unpredictable.

Theorem 8: the proposed IBS protocol is safe against impersonation attack.

Proof: when sub-PKG *S* has registered its identity with the PKG, the PKG generates the private key for *S.* The PKG and sub-PKG *S* are assumed to have a secure channel over which the private key is sent. Since an adversary *A* cannot compromise the secure channel between the PKG and the sub-PKG *S*, then it is unable to impersonate the sub-PGK because it cannot communicate with the PKG over that secure communication channel. This proof is valid for the mobile object impersonation as well.

4.3 Formal Verification Using Ban Logic

In this section the signing and verification phases (authentication process) of the proposed protocol is verified using BAN logic. The mobile object is referred to as MO and the verifier is referred to as V. The signature is ***σ***. Q_p is the verification key, and $Q_p{}^{*-1}$ is the private key corresponds to the public key $Q_p{}^*$.

Authentication process idealized messages:

- (M1) $V \rightarrow MO$: *e* (ignore this message since it does not contribute in the analysis.)
- (M2) $MO \rightarrow V$: $(Q_p, \{\boldsymbol{\sigma}\}Q_p{}^{*-1})$

Authentication process assumptions:

- (A1): $V \mid\equiv PK(Qp^*, MO)$
- (A2): $V \mid\equiv \#\, e$
- (A3): $V \mid\equiv MO \Longrightarrow \sigma$
- (A4): $V \mid\equiv II(MO)$
- (A5): $V \mid\equiv (\overset{Qp^*}{\longrightarrow} MO)$

Authentication process goals: the authentication process of the proposed protocol is based on signatures and the verification key. In order to authenticate the identity of the mobile object, the verifier must believe that the signature is true (correct) and that it has been sent by the mobile object. As a result, the following goals must be achieved.

- (G1): $GW \mid\equiv MO \mid\sim \sigma$
- (G2): $GW \mid\equiv MO \mid\mid\sim \sigma$
- (G3): $GW \mid\equiv \#\sigma$
- (G4): $GW \mid\equiv \sigma$
- (G5): $GW \mid\equiv MO \mid\equiv \sigma$

Authentication process analysis:

Step 1: by applying signing rule R14 on M2, A4, and A1, the first goal G1 is achieved. That is if V sees a message that is signed by the private key of MO, and V believes that $Q_p{}^*$ is a public key for MO, and it also believes that MO has a private key corresponds to $Q_p{}^*$, then V ought to believe that MO has said the message.

R14: $\frac{V \mid\equiv PK(Qp*,MO), V \mid\equiv II(MO), V \lhd \sigma(\sigma,MO)}{V \mid\equiv MO \mid\sim (\sigma)}$, so G1 $V \mid\equiv MO \mid\sim \sigma$ is achieved

Step 2: by applying synthetic rule R7 on A2, the third goal G3 is achieved. That is if V believes that part of the signature σ is fresh (which is *e*), then it believes that the signature σ is fresh.

R7: $\frac{V \mid\equiv \#(\sigma')}{V \mid\equiv \#(\sigma)}$, so G3 $V \mid\equiv \#(\sigma)$ is achieved.

Step 3: by applying the freshness rule R5 on G1 and G3, G2 is achieved. That is if V believes that σ is fresh and it also believes that MO has said σ, then V believes that MO has recently said σ.

R5: $\frac{V \mid\equiv \#(\sigma), V \mid\equiv MO \mid\sim (\sigma)}{V \mid\equiv MO \mid\mid\sim (\sigma)}$, so G2 $V \mid\equiv MO \mid\mid\sim (\sigma)$ is achieved.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 4: by applying the freshness rule R4 on G1 and G3, we can infer that V believes that MO believes σ (G5). That is if V believes σ is fresh and it also believes that MO said σ, then V believes that MO believes σ.

R4: $\frac{V \mid\equiv \#(\sigma), V \mid\equiv MO \mid\sim (\sigma)}{V \mid\equiv MO \mid\equiv (\sigma)}$, so G5 V $\mid\equiv$ MO$\mid \equiv (\sigma)$ is achieved.

Step 5: by applying jurisdiction rule R13 on A3 and G5, G4 is achieved. That is if V believes that MO controls σ and it also believes that MO believes σ, then V believes σ.

R13: $\frac{V \mid\equiv MO \Rightarrow \sigma,\ V \mid\equiv MO \mid\equiv \sigma}{V \mid\equiv \sigma}$, so G4 $V \mid \equiv \sigma$ is achieved.

5. COMPARISON AND EVALUATION

In this section, the proposed protocol is compared to other related protocols according to two factors. The first factor is key generation and distribution methods and the second factor is security attacks. Moreover, the performance of the proposed protocol is evaluated based on the computation cost of each phase.

5.1 Comparison and Discussion

Table 2 lists a comparison between the proposed protocols and its rivals according to the key generation and distribution methods. As illustrated in the table, all of the rival protocols, except [13], are based on bilinear pairing which is more complex than ECC [9]. Regarding key distribution, all protocols, including the proposed one, assume the use of secure channel.

Table 3 shows another comparison between the same protocols according to security attacks that protocols are safe against. One can notice that the proposed protocol is secure against all attacks except the anonymity attack since the identities of the mobile objects are transmitted over a public unsecure channel. The best rival protocol, i.e. [13], has counter measures against 4 out of the 7 considered security attacks. [14] is the only protocol that has counter measure against anonymity attack which is its sole counter measure.

Table 2: A Comparison between Related Protocols Methods According to Keys Generation and Distribution Methods.

Ref	Keys Generation Method	Keys Distribution Method
[5]	Bilinear Pairing	Secure Channel
[13]	ECC	Secure Channel
[15]	Bilinear Pairing	Secure Channel
[14]	Bilinear Pairing	Secure Channel
[16]	Bilinear Pairing	Secure Channel
Proposed	ECC	Secure Channel

Table 3: A Comparison between Related Protocols According to Security Analysis.

Attacks \ Ref	[5]	[13]	[15]	[14]	[16]	Proposed
Linking Signature	×	×	×	×	×	√
Impersonation	×	√	×	×	√	√
Replay Attack	×	√	×	×	×	√
Eavesdropping	×	×	×	×	√	√
Compromised object	×	√	×	×	×	√
Anonymity	×	×	×	√	×	×
Signature Forgery	×	√	√	×	×	√

5.2 Performance Evaluation

For each protocol, performance evaluation is performed by analyzing the computation cost according to the number of operations and the time needed to perform each operation that is used in various phases, see Table 4.

Evaluation results are shown in Table 5 which illustrates that the proposed solution is more efficient than other protocols since it does not use complex operations such as bilinear pairing, modular invers [11] [12].

The setup phase of the proposed protocol requires only one scalar multiplication in order to calculate the PKG public key Q_0. In sub-PKG setup phase, two scalar multiplication operations are needed to calculate U_s and h_s Q_0, one point addition operation to calculate Q_s, and one hash operation to calculate h_s. The extract phase requires same operations as in the sub-PKG setup phase. The sign phase requires one hash operation on the verifier side to calculate e, one scalar multiplication operations on the object side to calculate $Q_p = k \,.\, P$. Finally, the verification phase requires two scalar multiplication to calculate $.P$ and $e \,.\, O_{p*}$, in addition to one point addition operation to calculate $e \,.\, O_{p*} + Q_p$.

The number of operations and time needed to perform each operation for the rival protocols

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

have been calculated in the same way as for the proposed protocol. This calculation is based on the description of the details of each phase of these protocols as described in their corresponding references.

Table 4: Performance Metrics.

Metric	Description
N_{sm}	Number of scalar multiplication
N_h	Number of hash operations
N_{pa}	Number of point addition
N_{hp}	Number of hash to point operations
N_{pp}	Number of public points
N_{bp}	Number of bilinear pairing operations
N_{io}	Number of modular inverse operations
T_{sm}	Time to perform one scalar multiplication
T_h	Time to perform one hash operation
T_{pa}	Time to perform one point addition
T_{hp}	Time to perform one hash to point operation
T_{bp}	Time to perform one bilinear pairing operation
T_{io}	Time to perform one modular inverse operation

6. CONCLUSION AND FUTURE WORK

In this paper a hierarchical authentication protocol for IoT mobile objects is proposed. The proposed hierarchical architecture consists of three layers; the PKG layer, the sub-PKGs layer, and the objects layer. The proposed protocol is based on IBS to generate signing and verification keys. This protocol has five phases; PKG setup, sub-PKG setup, extract, sign, and verify.

The proposed protocol has been compared with other related protocols found in the literature. The comparison is done according to several factors which are: key distribution method, key generation method, and security attack model.

The proposed protocols can counter more attacks than its rivals. These attacks include linking signature, impersonation attack, replay attack, eavesdropping attack, compromised object, signature forgery attack. Anonymity is the only attack that is considered by one rival protocol but not the proposed one.

In addition to that, quantitative measure for performance evaluation has been included. The evaluation shows that the proposed protocol outperforms other protocols in terms of the total computation cost since it does not use expensive hash to point operation, modular inverse operation, and bilinear pairing operation.

As future work, the proposed protocol could be enhanced to take into account other attacks such as anonymity among others.

REFERENCES

[1] S. Li, L. Da Xu and S. Zhao, "The Internet Of Things: A Survey," *Information Systems Frontiers,* vol. 17, no. 2, 2015, pp. 243–259.

[2] W. Almobaideen, M. Allan and M. Saadeh, "Smart Archaeological Tourism: Contention, Convenience And Accessibility In The Context Of Cloud-Centric IOT," *Mediterranean Archaeology and Archaeometry,* vol. 16, no. 1, 2016, pp. 227-236.

[3] W. Almobaideen, M. Saadeh, N. Al-Anbaki, R. Zaghloul and A. Aladwan, "Geographical Route Selection Based On User Public Transportation and Service Preferences," in *9th International Conference on Next Generation Mobile Apps, Services and Technologies (NGMAST)*, Cambridge, 2015.

[4] M. Saadeh, . A. Sleit, M. Qatawneh and . W. Almobaideen, "Authentication Techniques for the Internet of Things: A Survey," in *Cybersecurity and Cyberforensics Conference*, Amman, Jordan, 2016.

[5] H. Li, Y. Dai, L. Tian and H. Yang, "Identity-Based Authentication for Cloud Computing," in *IEEE International Conference on Cloud Computing*, 2009.

[6] A. Shamir, "Identity-based cryptosystems and signature schemes," *In Advances in Cryptology CRYPTO '84, of LNCS,* vol. 196, 1984, pp. 47-53.

[7] Girish and P. H.D , "Identity-Based Cryptography and Comparison with traditional Public key Encryption: A Survey," *International Journal of Computer Science and Information Technologies,* vol. 5, no. 4, 2014, pp. 5521-5525.

[8] L. Yan, C. Rong and G. Zhao, "Strengthen Cloud Computing Security with Federal Identity Management Using Hierarchical Identity-Based Cryptography," in *IEEE International Conference on Cloud Computing*, China, 2009.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[9] S. Biswas and J. Miˇsi´c,, "A Cross-layer Approach to Privacy-preserving Authentication in WAVE-enabled VANETs," *IEEE Transactions on Vehicular Technology,* vol. 62, no. 5, 2013, pp. 2182 - 2192,.

[10] M. Burrows and M. Abadi, "A Logic of Authentication," *ACM Transactions on Computer Systems,* vol. 8, no. 1, 1990, pp. 18-36.

[11] C.-H. Tsai and P.-C. Su, "An ECC-Based Blind Signcryption Scheme for Multiple Digital Documents," *Security and Communication Networks,* vol. 2017, p. 14.

[12] N. Tahat and E. E. Abdallah, "A new signing algorithm based on elliptic curve discrete logarithms and quadratic residue problems," *Italian Journal of Pure and AppliedMathematics,* vol. 32, 2014, pp. 125–132.

[13] S. Biswas, J. Miˇsi´ and V. Miˇsi´, "An identity-based authentication scheme for safety messages in WAVE-enabled VANETs," *International Journal of Parallel Emergent and Distributed Systems,* vol. 27, no. 6, 2012, pp. 541-562.

[14] Y. Zhang, . L. Yang and . S. Wang , "An Efficient Identity-Based Signature Scheme for Vehicular Communications," in *Computational Intelligence and Security (CIS)*, Shenzhen, China, 2015.

[15] D. He, N. Kumar, K.-K. R. Choo and W. Wu, "Efficient Hierarchical Identity-Based Signature With Batch Verification for Automatic Dependent Surveillance-Broadcast System," *IEEE Transactions on Information Forensics and Security,* vol. 12, no. 2, 2017, pp. 454 - 464.

[16] A. Qousini, "Role-Based Access Control Model for Privacy Preservation in Cloud Computing Environment," The University of Jordan, Amman, 2015.

[17] A. Corbellini, "Elliptic Curve Cryptography: a gentle introduction," 2015. [Online]. Available: http://andrea.corbellini.name/2015/05/17/elliptic-curve-cryptography-a-gentle-introduction/. [Accessed 18 8 2017].

[18] L. Butty´an, S. Staamann and U. Wilhelm, "A Simple Logic for Authentication Protocol Design," in *Proceedings of 11th IEEE Computer Security Foundations Workshop*, USA, 1998.

[19] J. Wessels, "Applications of BAN-Logic," 2001.

[20] S. H. Islam and . G. Biswas, "A pairing-free identity-based two-party authenticated key agreement protocol for secure and efficient communication," *Journal of King Saud University - Computer and Information Sciences,* vol. 29, no. 1, January 2017, pp. 63–73,

[21] A. Menezes, "Evaluation of Security Level of Cryptography: The Elliptic Curve Discrete Logarithm Problem (ECDLP)," University of Waterloo, 2001.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

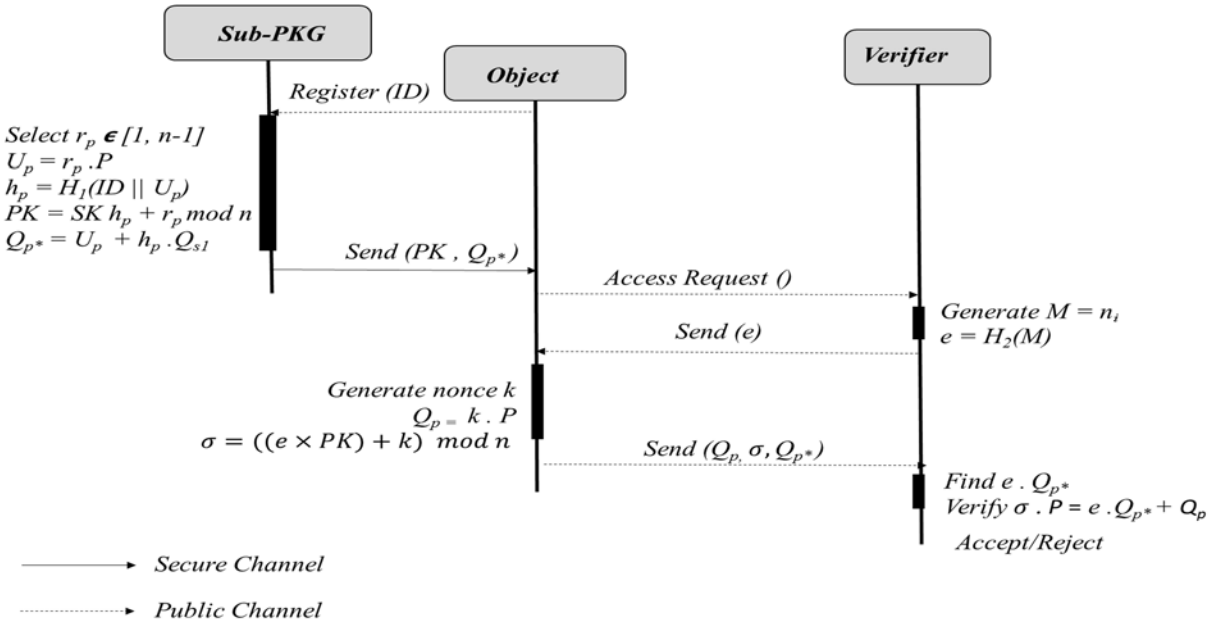

Figure 2: The Extract, Signing, and Verification Phases.

*Table 5: Performance Evaluation of Each Phase. * L is the hierarchy level, assumed to be 3.*

Phase	**Ref / Metric**	[5]	[13]	[15]	[14]	[16]	**Proposed**
Setup	N_{sm}	1	3	1	1	1	1
Sub-PKG Setup	N_{io}	0	1	0	0	0	0
	N_{sm}	2	0	2	3	L	2
	N_h	0	1	1	0	0	1
	N_{pa}	1	0	0	0	1	1
Extract	N_{io}	0	0	0	1	0	0
	N_{sm}	4	0	2	2	L	2
	N_h	0	0	1	6	0	1
	N_{pa}	2	0	1	0	1	1
Sign	N_{io}	0	1	0	0	0	0
	N_{sm}	1	1	2	2	1	1
	N_h	0	2	1	1	1	1
	N_{pa}	1	0	1	0	1	0
Verify	N_{io}	0	1	0	0	0	0
	N_{sm}	1	3	3	1	L	2
	N_h	0	2	3	1	1	0
	N_{pa}	0	2	3	0	0	1
All phases	N_{hp}	4	0	0	4	1	0
All phases	N_{pp}	5	3	7	7	L +1	3
All phases	N_{bp}	4	0	2	2	L + 3	0
All phases	*Total*	$9T_{sm} + 4T_{pa} + 4\ T_{hp} + 4\ T_{bp}$	$7T_{sm} + 5T_h + 2\ T_{pa} + 3T_{io}$	$10T_{sm} + 6T_h + 5T_{pa} + 2\ T_{bp}$	$9T_{sm} + 8T_h + T_{io} + 4\ T_{hp} + 2\ T_{bp}$	$11T_{sm} + 2T_h + 3T_{pa} + T_{hp} + 6T_{bp}$	$8T_{sm} + 3T_h + 3\ T_{pa}$