Journal of Advanced Computer Science & Technology, 4 (x) (2015) xxx-xxx
www.sciencepubco.com/index.php/JACST

doi:

A Novel Authentication and Access control framework in Wireless Sensor Networks

Babak Nouri-Moghaddam[1]*, Hamid Reza Naji[2]

[1]Department of Computer Engineering, Iran University of Science & Technology, Iran, Tehran
[2] Department of Electrical and Computer Engineering, Graduate University of Advanced Technology, Iran, Kerman
**Corresponding author E-mail:*
B_NouriMoghaddam@ind.iust.ac.ir
Babaknouriit85@gmail.com

Abstract

Wireless Sensor Networking continues to evolve as one of the most challenging research areas. Considering the insecure nature of these networks and the fact that sensor nodes are distributed in a hostile environment, having a well-implemented security scheme is absolutely essential. Bearing in mind the important security services like authentication and access control, we have proposed a novel security framework for these networks. The new framework is based on Kerberos authentication and access control system. The Kerberos has been adopted for WSNs by utilizing Bloom Filter data structure and Elliptic Curve cryptography. In the proposed scheme, Bloom Filter data structure is used in a novel way; we have used this data structure to get rid of Public Key's certificates. By combining Bloom Filter data structure and Elliptic Curve cryptography, we achieved a very light robust security framework that offers Authentication, Access Control, and key sharing services. The analysis results showed that our scheme provides more security services and is more robust in the presence of attacks compared to the previous schemes. In contrast, simulation results indicated that our system had significant improvements over the other schemes in many aspects such as power and time expenditure.

Keywords: *Wireless Sensor Network; Authentication; Access Control; Kerberos; Bloom Filter; Elliptic Curve Deffie-Hellman.*

1. Introduction

Following the recent advances in micro-electromechanical systems (MEMS) technology, wireless communications, and digital electronics, it is technically and economically practical to manufacture a large number of small and low-cost sensors [1]. Each tiny sensor node includes a processor, a sensing unit, a communication device, and a power supply unit. The Wireless Sensor Networks (WSNs) are formed by distributing a large number of sensor nodes in the environment and assigning a base station to the area. These networks have a wide variety of applications such as military, environmental, healthcare, home and commercial applications [2].

Because of the WSN's wide range of applications, the sensor nodes in these networks are usually distributed in hostile environments, necessitating security services for these networks. WSNs have particular characteristics (e.g., Power limitation, insecure nature of the wireless communication, limited computational power, etc.) which make them unique in many aspects. Hence, most of the traditional approaches (e.g. security approaches in ad-hoc networks, etc.) cannot be applied to these

networks [1, 3-5]. For example, in military applications, sensor nodes are used for gathering data from the battlefield where there always is a possibility of tapping on communications and even capturing sensor nodes. Considering these facts, there ought to be mechanisms to establish security services with the respect to the constraints of WSNs.

Security services cover Integrity, Confidentiality, Authentication, Access Control, and Non-repudiation. These services are designed as a countermeasure against security attacks. However, the issues raised in the references [1, 6-8] show that Authentication service is one of the most important services in WSNs, thereby establishing this service is a keystone for having other services to secure the network [4, 9, 10]. Access Control services are necessary when we have users with several levels of access permissions [10, 11].

Most of the proposed schemes fall into two categories by the type of encryption they use: (1) Symmetric key cryptography and (2) Public key cryptography. In WSNs, due to the strict resource limitation of sensor nodes, symmetric key cryptographies showed promising feature but after implementation of these methods, the problems like insufficient scalability and vulnerability to physical attacks emerged [12]. Meanwhile, with new advances in microprocessor technologies, the public key cryptography has been welcomed, though communicationally and computationally expensive on the sensor nodes. Therefore, novel approaches are attempting to merge both of these methods to yield hybrid approaches.

Due to the variety of security attacks, designing a comprehensive and robust security scheme for WSNs is essential. So far many solutions considering WSNs' specification have been suggested, with few of which having enough integrity and robustness [5, 9-11, 13-16].

In this paper, we have offered a comprehensive Authentication and Access Control framework for wireless sensor networks. The idea of our approach is based upon the Kerberos Authentication and Access Control mechanism. Like the Kerberos, we have considered two nodes in each cluster as authentication and ticket-granting server to provide comprehensive Authentication and Access Control framework. As a result, we have separated the authentication and Access Control overhead from each other and assigned them to different nodes. For Authenticating and key sharing phase, we utilize the Bloom Filter (BF) data structure to forward security parameters in a novel and confident way. In addition, this new method allows us to eliminate the requirement of public key's certificates, providing us with a low-cost and fast comprehensive authentication and key exchange service. To achieve this, we use a combination of BF data structure and Elliptic Curve Cryptography (ECC). The BF data structure is a space efficient data structure using k hash function to store set $E = e_1, e_2, ..., e_m$, in a vector V with size of n, where $n \leq m$. Furthermore, the cost of lookup and insertion operation is $O(k)$ [17, 18]. Because of the nature of one-way hash functions, it is impossible to get any information about the original data by having vector V. For authenticating a new User/Sensor node, one should concatenate its password and public key with some other important information to create *BF* vector V. The vector V is sent to the Authentication Node (Auth-Node) and the identity of the new User/Sensor node will be verified. If the new node's identity and provided information are valid, it will be recognized as a valid user/sensor node.

For mutual authentication, Auth-Node will build vector V' with new user/sensor node's information, known only to Auth-Node and user/sensor node. On top of that, Auth-Node will couple its public key to applicant's information and send the V' to new user/sensor node. If the vector V' is valid, the mutual authentication will be successful. Owing to this new approach, if the new user/node is a valid node, then all information, which is contained in the vector V, will be valid too. In addition, because of the nature of one-way hash function and secret password, it is impossible to forge vector V. As a result, the public keys will be authenticated without the need for public keys' certificates.

We implement Elliptic Curve Deffie-Hellman (ECDH) key exchange scheme to share a symmetric key among parties, which is necessary to have a low-cost confidentiality in future communication.

ECC is a well-known public key algorithm that has some features like small key size and lower computational power requirement, which is very promising in WSNs. The strength of this algorithm is because of the difficulty of solving Elliptic Curve Discrete Logarithm Problem (ECDLP). After authentication phase, both sides' public keys are authenticated and for key sharing by applying the ECDH algorithm [19, 20], each party will compute the shared symmetric key. From this point on, they are able to use this shared symmetric key to secure their communications using low-cost symmetric key methods.

In Access controlling phase, the Auth-Node sends the ticket request on behalf of the new user/node to the Ticket Granting Server Node (TGS-Node). TGS-Node checks the access table and issues an encrypted ticket for the new user/node. This ticket uses the BF to issue vector *T* as a ticket, which is encrypted to have high security.

Evaluations showed that our framework offers good Authentication and Access control security services. As result, the new approach is well protected against most attacks. Because of the innovative way of utilizing Bloom Filter and ECDH, the energy and communication overhead cost is less than most of other protocols.

The rest of the paper is organized as follow. Section II briefly reviews related works. Background of ECC and BF data structure which constitute the basis of the proposed method are described in Section III. The proposed security scheme is presented in Section IV. Section V presents the security analysis and performance evaluation of our method. Finally, Section VI concludes the paper and outlines the investigations for the future works.

2. Related Works

Most of the proposed schemes fall into two categories by the type of encryption they used: (1) Symmetric key cryptography and (2) Public key cryptography. In the beginnings of WSNs due to the strict resource limitation of sensor nodes, symmetric key cryptography showed promising feature, but after implementation of these methods, the problems like insufficient scalability and weakness against physical attacks emerged. Meanwhile, with new advances in microprocessor technologies, the public key cryptography has been welcomed; however, these methods are very expensive on the sensor nodes. Therefore, novel approaches endeavor to merge both of these methods to achieve hybrid approaches.

Researchers have been examining the implementation of Authentication and Access Control services in wireless sensor networks from different angles like Broadcast Authentication, Authentication of new nodes, user Authentication, user Access Control, and development of encryption algorithms.

μ-TESLA authentication scheme is one of the earliest approaches which covers minimum security standards[21]. This scheme is a part of the SPINS security protocol. Symmetric Key cryptography is keystone of μ-TESLA scheme. In the initial phase, base station defines a key disclosure time interval. Moreover, base station generates one-way hash key chain and assigns a key to each interval. During each interval, base station encrypts all messages with the interval's key. After broadcasting encrypted message, base station reveals the interval key to all the nodes within the network. Sensor nodes use this key to authenticate the source of the incoming messages. The major drawback of μ-TESLA scheme is its vulnerability to the Denial-Of-Service (DoS) attacks. In addition, this scheme needs loosely timed synchronization between sensor nodes and base station, which is such a challenging task in order to meet in WSNs.

In 2012, Liu and colleagues[9] presented a novel scheme based on the public key Cryptography (PKC), and designed this scheme for broadcast message authentication. They used ECDSA as the encryption algorithm. In their scheme, *N* messages are denoted by a vector *M* (n=kb, *k* is an integer

multiplier) and they are partitioned into k blocks (EB_i, $i = 0, \ldots, k$) and Sender computes Hash-based Message Authentication Code (HMAC) value for each group. Fig. 1 shows the example of this scheme in practice. A random value is assigned to part d_{k+1} of the group EB_k , and then the sender computes the HMAC value for EB_{k-1} . The computed HMAC value is assigned to the EB_{k-2} , and the same as before, each group's HMAC value will be attributed to the previous group up to the EB_0. For EB_0, sender signs d_1 with its private key and sends EB_0 . Upon receiving EB_0, the receiver checks the sender's signature using the sender's public key. If it is a valid signature, then the message source will be authenticated. From here on, the receiver can authenticate each group of the *EB*s by computing the HAMC value and comparing it to the previous group's *d* value. Determining the number of groups and number of messages in each group depends on the network specifications. Furthermore, using public key certificates and digital signatures are costly operations on the WSNs.

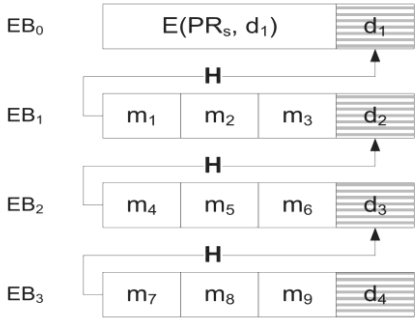

Fig. 1. Example of PKC in practice

Identity-based Multi-user Broadcast Authentication System (IMBAS) is another broadcast authentication approach put forward by Cao et al. [4]. They implemented the combination of ECC and Identity Base approach to verify the sender of broadcast messages. Although most of the public key methods use the certificates for authenticating the owner of the public key, e.g. RSA, ECC, the Identity-based approach uses the owner's ID as its public key certificate. Therefore, this approach has less communication overhead than the others. By combining these two approaches, the communication overhead will be reduced, but using digital signatures and Identity-Base methods significantly increases the computation overhead of IMBAS, which has a great impact on network lifetime. IMBAS has covered more security services than its predecessors, except for Access Control service which is critical for network security. $\varepsilon IBAS$ is another Identity-Based approach for broadcast message authentication proposed by Shim et al.[22] This approach makes use of optimal Identity-Base operation to reduce Identity-Base operation cost. Compared to the symmetric key methods and ECC, the pairing operation in Identity-Base method is so expensive on sensors. As a result, this approach suffers from high computation overhead and is impractical to use in WSNs.

In 2012, Al-Mahmud and his colleagues presented new scheme to authenticate sensor nodes in WSNs[23]. They applied ECC and digital signature to address this issue. In their scheme, all nodes have their unique pair of public and private keys, assigned by base station. Nodes which want to join the network must send an authentication request to their neighbors signed by their private keys. Upon receiving the request, the receiving node will confirm the signature validity. Although this approach has proved to deliver a minimum overhead on the network by using ECDH, however, employing digital signature results in greater overhead over sensor nodes. Additionally, they used digital signature for Access Controlling, which accounted for the poor performance of this approach.

Kumar Das et al. [24] presented a new approach for authenticating WSNs' users who can access the sensors' data locally. In their approach, they have used smart cards to handle the login operation. This approach has promising features like using symmetric key and dynamic password change; however, their scheme does not cover any Access Control mechanism. Besides, the entire

authentication operation will go through the base station, making the network vulnerable to attacks like DoS.

Wang et al. [25] (HBQ scheme) applied public key cryptography based on ECC to solve the problem of symmetric key approaches in terms of scalability, key storage, and key pre-distribution. Nevertheless, the performance evaluation has shown that HBQ is still burdensome for sensors, and leads to the impracticability of implementation. Le et al. [26] (ENABLE scheme) have solved security limitations and performance issues in HBQ. However, it depends on a trusted third party (e.g. Key Distribution Scheme (KDC)) to deal with the significant ECC operations. Communicating with an on-line KDC always introduces significant cost increase in healthcare. Furthermore, failure of KDC may result in failure of the security function for the network.

Le et al. [11] have proposed a new approach named MAACE that tries to solve the ENABLE shortcomings. They develop the idea of using another layer of nodes in the network called coordinate nodes responsible for authentication on behalf of the base station. This approach uses public key certificates for access controlling that causes heavy load on the nodes communication and computation. Nouri et al. [27] presented another improvement in HBQ. In their scheme, they tried to overcome the high computation overhead of HBQ by using Bloom Filter as pre-Authentication. However, their proposed approach has some shortcomings like number of supported services and scalability.

Overall, none of the security protocols resolve the important issues in WSNs security e.g. Access control, key distribution, confidentiality, etc. completely. We have introduced a new approach that supports the most important security services as well as providing customized security services for WSNs' specifications.

3. Building Blocks

3.1. Elliptic Curve Cryptography

The use of elliptic curves in cryptography was suggested independently by Neal Koblitz and Victor S. Miller in 1985 [28, 29]. Recently, ECC has attracted much attention as the security solutions for WSNs because of the small key size and low computational overhead. For example, 160-bit ECC offers the comparable security to 1024-bit RSA[1][9, 30]. It is based on the algebraic structure of elliptic curves over finite fields. On the other hand, ECC multiplication operation feasible on sensor nodes. It takes only 0.81 second on 8-bit CPU Atmel ATmega-128 at 8 MHz [11, 19].

An elliptic curve consists of the points satisfying the equation:

$$y^2 + axy + by = x^3 + cx^2 + dx + e \quad (1)$$

The special case of this equation over a Galois Field of order p , is presented as follows:

$$y^2 = x^3 + ax + b \quad (2)$$

Where x, y, a and b are elements in $GF(q)$ (a Galois Field of order, and p is a prime). Each choice of $(a\ ,\ b)$ yields a different elliptic curve and they ought to abide by this rule:

$$4a^3 + 27b^2 \bmod p \neq 0 \quad (3)$$

For example, Figure 2 shows an elliptic curve of $y^2 = x^3 - 2x + 1$.

[1] RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described the algorithm in 1977.

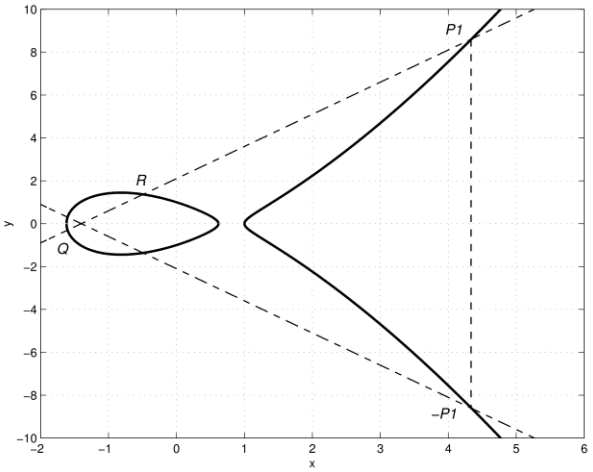

Fig. 2. Elliptic curve of $y^2 = x^3 - 2x + 1$

The elliptic curve group operation is closed under addition so that the addition of any two points is also a point in the group. Let the points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be in the elliptic group $E_p(a,b)$ and O be the point at the infinity. The rules for addition over the elliptic group $E_p(a,b)$ are:

1. $P + Q = Q + P$ 1.
2. If $x_2 = x_1$ and $y_2 = -y_1$, that is $P = (x_1, y_1)$ and $Q = (x_2, y_2) = (x_1, -y_1) = -P$, then $P + Q = O$.
3. If $Q \neq -P$, then their sum $P + Q = (x_3, y_3)$ is given by:

$$x_3 = \lambda^2 - x_1 - x_2 \bmod p \tag{4}$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \bmod p \tag{5}$$

Where λ is:

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & if\ P \neq Q \\ \dfrac{3x_1^2 - a}{2y_1} & if\ P = Q \end{cases} \tag{6}$$

3.2. Elliptic Curve Diffie–Hellman (ECDH)

The original Diffie-Hellman secret sharing protocol (Diffie and Hellman, 1976 [31]) requires a key of at least 1024 bits to achieve sufficient security. Unfortunately, low-power architecture, such as MSP430 and ATMega-128 cannot afford the large memory overhead. Diffie–Hellman scheme is based on ECC, and it can achieve the same security level as 1024 bit key RSA, with only 160-bit key size [9, 30]. A typical ECDH scheme is shown in

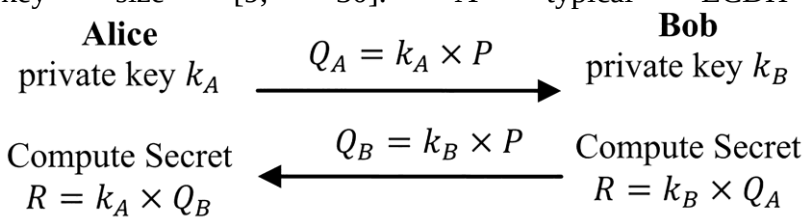

Fig. **3**. Initially, Alice and Bob agree on system base point *P* and generate their own public key Q_A and Q_B . To share a secret, Alice and Bob exchange their public keys. After that, Alice multiplies its private key with Q_B and the resulting point R will be the secret. Eve, an eavesdropper, may overhear the communication and learn the public keys from Alice and Bob. However, with the knowledge of P, Q_A and Q_B , it is computationally impossible for Eve to get Alice and Bob's private keys. As a result, she cannot figure out the secret R.

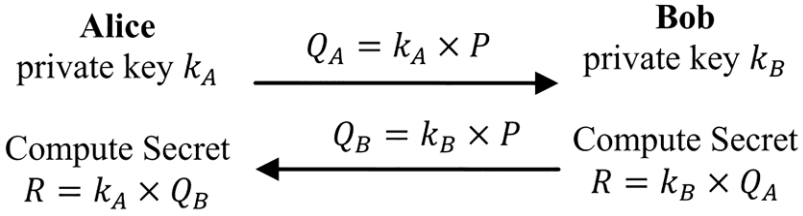

Fig. 3. ECDH method in practice

3.3. Bloom Filter

The bloom Filter data structure is designed to do a quick lookup. For this reason, it can perform lookup in order of $O(k)$, where *k* shows the number of hash functions. Additionally, this data structure is extremely space efficient. To display a set $E = e_1, e_2, ..., e_n$ with *n* members, we consider *V* as a *BF* vector with *m* bits length. In the initial state, all bits are set to zero. In addition, there should be *k* one-way hash functions $h_1, h_2, \ldots h_k$, the output of which should be in [*0, m-1*]. Each member of the set *E* will be hashed by *k* hash function and the results determine the block number in the vector *V,* and then the pointed block will be set to one. In the lookup phase, to check whether *x* is a member of the *E* series or not, the vector V' will be calculated for input *x* using the hash functions, if $V' \subseteq V$ than the *x* is a member of the *E* set.

The Bloom Filter data structure has a positive error for checking membership of x in the set E. This error is calculated from equation below:

$$f = (1 - e^{\frac{-kn}{m}})^k \tag{7}$$

4. The Proposed Idea

In this section, the fundamental features and properties of our work are explained. Our proposed scheme has four main phases for initialization, establishing authentication, access control, and confidentiality. For registration during the pre-distribution and network organization, we have considered Network initialization phase as phase one. Each node/user in the network who wants to gain access to the network must go through all four main phases. Along with the main phases, we have considered three optional phases for dynamic password change, authentication forwarding, and node/user revoking. These three phases are designed to provide more security services, which results in more robust security framework.

The four main phases are:

1) Network Initialization
2) Authentication and key sharing
3) Access controlling and ticket issuing

4) Accessing the network

Next subsection explains each phase in detail.

4.1. Network Initialization:

In this phase, each node/user is registered in the Base Station (BS), and gets security parameters like [*KU, KR*] pairs, hash set, *BF* vector size, etc. There is a difference between user registration and node registration in the Initialization phase, where users initiate registration phase; however, for nodes, this phase starts with a BS. The difference is caused by the fact that users always apply for registration, but sensor nodes do not possess this ability. User registration is also explained in this section. The user initiates the registration phase by forwarding the registration request to BS. BS assigns an ID_{user} for the user and asks him/her to enter a password (PW). The user enters its *PW* and sends it to BS via a secure channel. On receiving the *PW*, BS computes the [*KU, KR*] for the user and forwards them along with network security parameters to the user. Additionally, BS inserts user's ID_{user}, *PW*, and [*KU, KR*] into its main table. For future authentication, BS sends the node's user ID_{user} and *PW* to Auth-Nodes. In addition, BS forwards user's ID_{user} and access vector to TGS-Node. Fig. 4 shows the message exchange sequence of this phase.

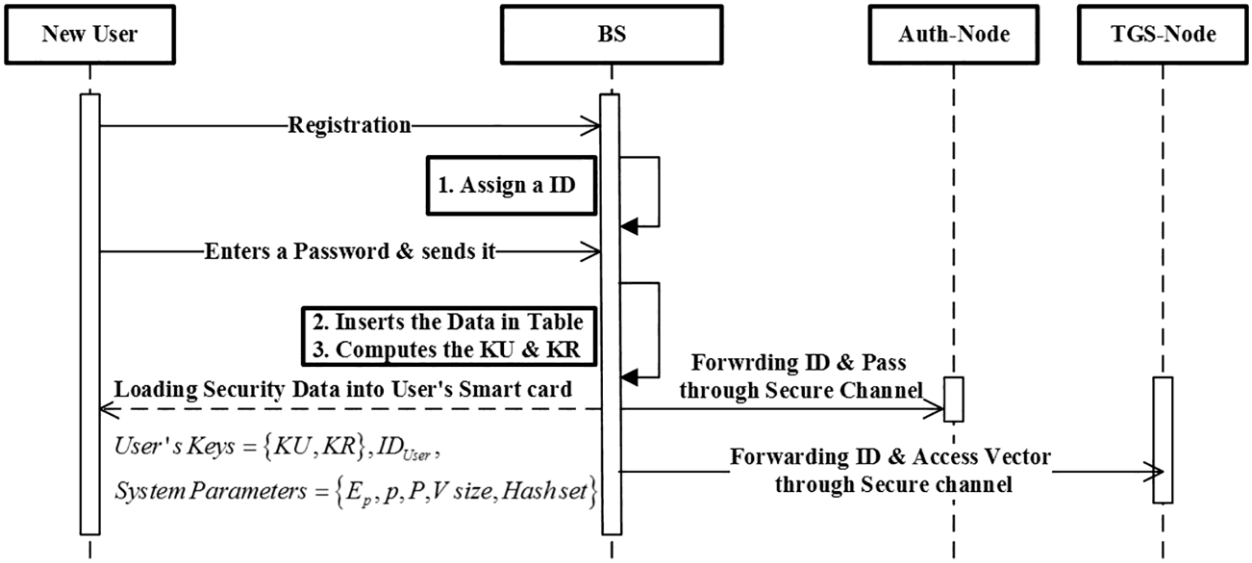

Fig. 4. Message exchange sequence in Network Initialization phase

4.2. Authentication and key sharing

After deploying the nodes and organizing the network in each cluster, Auth-Node starts to broadcast a beacon message that contains Auth-Node's ID, TGS-Node's ID, and timestamp (TS_1). This message lets the users and nodes learn about Auth-Node and TGS-Node. Upon receiving the beacon, the user/node sends an authentication request to Auth-Node that contains the user's ID_{new}, TGS-Node ID_{tgs}, Auth-Node ID_{Auth}, TS_2, *BF* vector, and user's KU_{new}. To generate the *BF* vector V, the user inserts data combination of the user's ID_{new}, TGS-Node ID_{tgs}, user's *PW*, TS_1, and user's KU_{new} ($[ID_{new} \,|\, ID_{tgs} \,|\, PW \,|\, TS_1 \,|\, KU_{new}]$). Due to the nature of the one-way hash functions, no one can trace *BF* vector back to its original input data. Thus, the password remains confidential. Auth-Node receives the request and checks user's ID; if it is a valid *ID*, then step 2 starts. In this step, Auth-Node builds the *BF* Vector V' with received data and user's *PW*, which is only known to the users, and Auth-Nodes. If V' identical to V , then the user is authenticated. Because of the one-way hash function, it is impossible to forge the *BF* vector; besides, the *BF* vector contains TS_1 that will be changed by Auth-Node in next beacon broadcasting. As a matter of fact, TS_2 is embedded in *BF*

vector and if Auth-Node detects a duplicate TS_2, it will discard the second request. As a result, if anyone gains access to *BF* vector, he cannot use it at any time afterward.

We have considered mutual authentication step to provide more reliability. Once the user/node is validated, Auth-Node forwards the reply message to user/node, including *ID* information, a *BF* vector, and Auth-Node's KU_{Auth}. To generate the *BF* vector, Auth-Node uses user/node's *PW*, TS_2, and Auth-Node's KU_{Auth}. Since no one other than the user and Auth-Node is aware of the user's *PW*, it is impossible to generate valid vector without knowledge about user's *PW*. On receiving Authentication reply, the user/node rebuilds the *BF* vector by received data and its *PW*, and if it is equal to the received one, then Auth-Node is validated too.

When all parties' identity is confirmed, we consider the key sharing method using ECDH. The novel idea about using public key sharing scheme in a light way lies in this part. During authentication phase, each party sends its public key in plain text and embedded form in the *BF* vectors. Because it is impossible to forge *BF* vector to include fake public key when one party's identity is verified, the authenticity of its public key will be approved too. Due to this fact, public keys will be authenticated without employing any certification or certificate authority. For key sharing, each party should perform ECDH via other party's *KU*. After this, both parties will hold the same symmetric key and from here, they can have secure communication with lightweight symmetric key cryptography. The detailed sequence is presented in Fig. **5**.

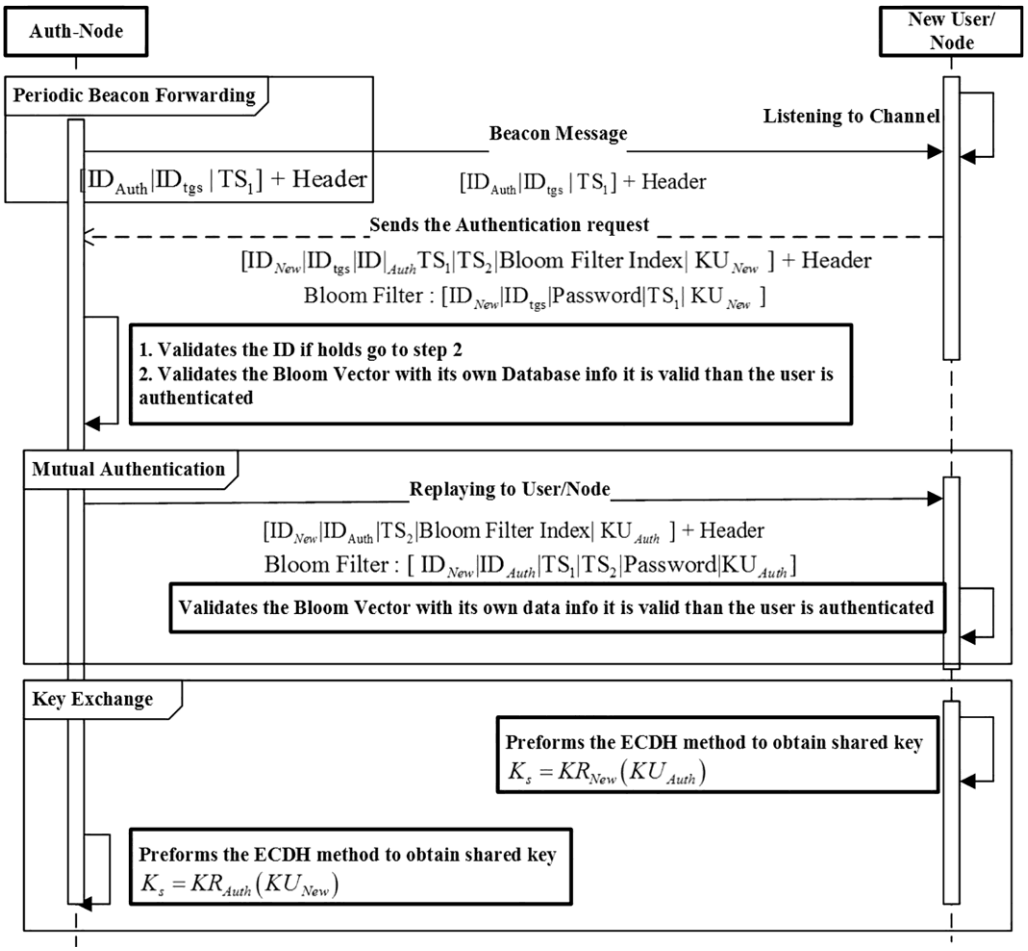

Fig. 5. Sequence diagram of the Authentication and key sharing phase

4.3. Access controlling and Ticket issuing:

After mutual authentication and key sharing phase, we should check for user's access level , then issue a ticket according to their access level. For this purpose, we designed Access controlling and

ticket issuing phase. This phase is started by Auth-Node requesting ticket for newly authenticated user/node. Auth-Node sends ticket request to TGS-Node. The request is encrypted by their secret shared key. Ticket request contains user/node's ID_{new} and the shared key Ks_{new} between Auth-Node and user/node. Having received the request, TGS-Node checks the access privilege table for valid user and access level; if there is a result, TGS-Node issues a ticket to the new user/node. To have efficiency in memory usage and communication overhead, we have utilized the *BF* data structure to issue a ticket. The ticket includes a *BF* vector which contains access privilege, Lifetime, and some other parameters. To have robust security against sniffing and man in the middle attacks, we embedded Lifetime to tickets. Hence, after Lifetime expires, the ticket will be invalid to use in the network. The TGS-Node encrypts[1] the ticket with its own symmetric key to protecting the ticket from any future changes. The encrypted ticket will be encrypted once more with the user/node symmetric shared key to protect it against spoofing attack. In the last step, TGS-node forwards the encrypted ticket to the user/node. After the new user/node receives the TGS-node message, user/node will decrypt the message with its shared key and extract the ticket. Fig. **6** shows the message exchange sequence of this phase.

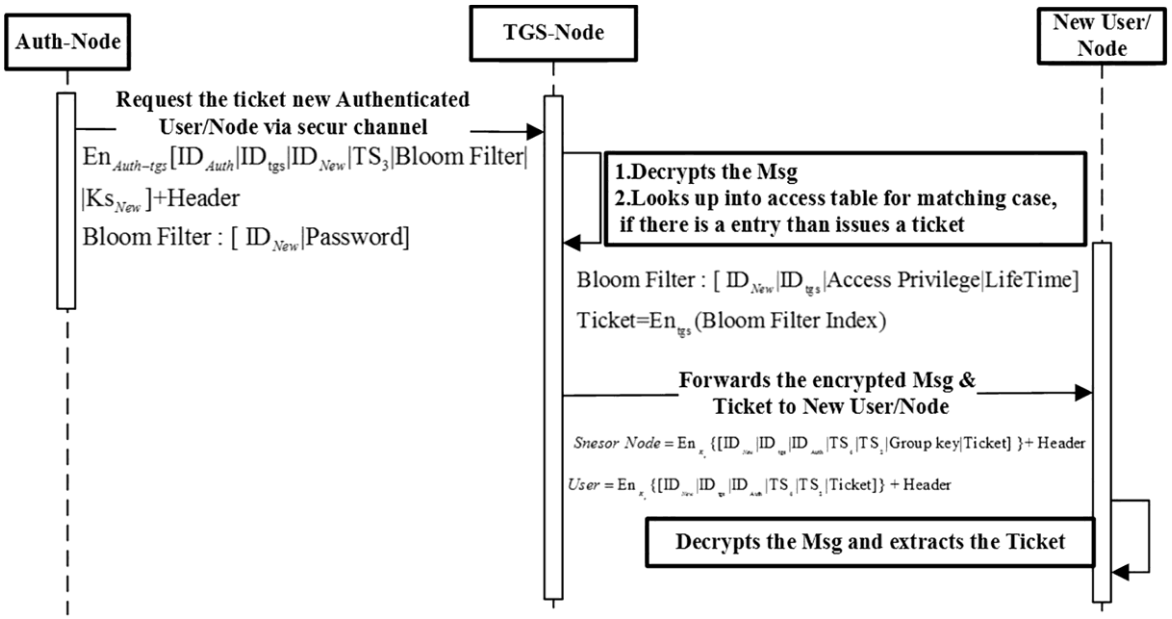

Fig. 6. Message exchange sequence in Access controlling and ticket issuing phase

4.4. Accessing the network:

When a user/node receives its ticket, it can access any entity in the network. For gaining access to the network, users/nodes should go through following phase. The user/node initiates this phase by sending access request to the Dest-Node and generating a message that contains some *ID* information and its encrypted ticket. Dest-Node receives the access request message from User/Node and checks its *TS*; if the *TS* is fresh, then it transmits the user/node's ticket to TGS-Node for verification. Dest-Node encrypts the verification request with its shared symmetric key and forwards it to the TGS-Node. By receiving the verification request, TGS-Node decrypts the message and extracts the ticket. Having extracted the ticket, TGS-Node checks the user/node's identity. If the provided information holds valid, TGS-Node generates a session key for Dest-Node and user/node. The session key and acknowledgment will be encrypted and forwarded to Dest-Node. Consequently, Dest-Node sends acknowledgement message to the user/node. In the final step, the User/Node generates session key using HMAC method. This session key is identical to the session key created by TGS-Node. Because the user and TGS-Node have knowledge about the number of

[1] $En_{key's\ owner}[parameters]$: means that the parameters encrypted with owners shared key.

successful accesses, they can perform HMAC on the same information and produce the same session key independently. The message exchange sequence of this phase is shown in Fig. 7.

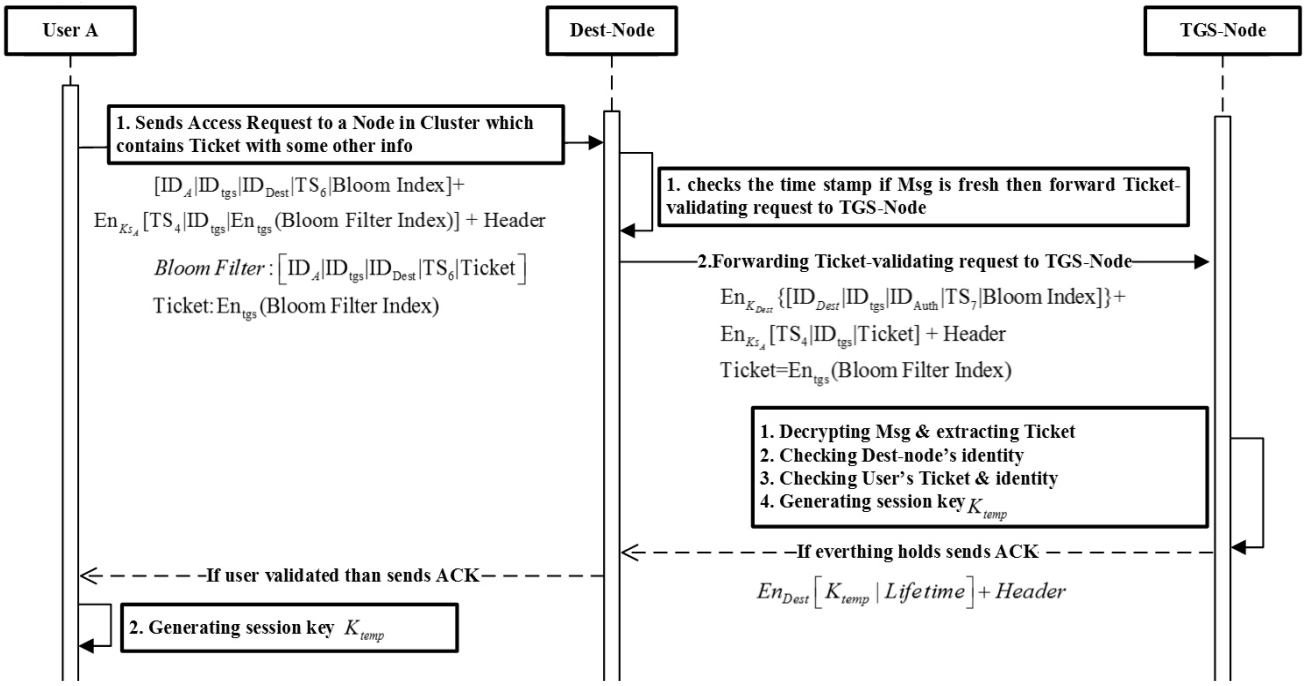

Fig. 7. Accessing the network phase in detail

4.5. Optional Phases

As mentioned before, we have considered some optional phases to provide more security services and robustness against different attacks. These optional phases include dynamic password change, authentication forwarding, and node/user revoking. Dynamic password change has been designed for users to change their PW locally and freely in a secure manner. When a user/node is a mobile entity and wants to join another cluster, it can use Authentication forwarding phase by which there will be no need for fresh authentication in the new cluster. Only the user/node's ticket should be exchanged between two clusters TGS-Node. Therefore, there is great save on energy for user/clients. In addition, when we want to remove a user/node from the network, we can use user/node revocation protocol.

For brevity, we explain only the dynamic password change phase in detail. When the user wants to change its PW, he/she enters its old PW and new PW, and then he/she encrypts them with his share symmetric key. The user sends PW change request to the Auth-Node. When Auth-Node receives this request, it decrypts the message, and verifies the old PW. If it is valid, it will update its table with user's new PW. In next section, we examine our proposed scheme in action and explain the results.

5. Security analysis and performance evaluation

This section presents security analysis and simulation results of the proposed scheme.

5.1. Security analysis

In this section, the advantages of our scheme in security perspective are explicated. The new scheme provides many security services which make it more robust. The services provided by our scheme are mutual authentication, key sharing, access controlling, confidentiality, non-repudiation, data integrity, dynamic password change and user/node revocation. In terms of providing security

services, our scheme provides a variety of security services and is impenetrable against most common attacks in WSNs. We now show that our scheme can resist the following attacks.

5.1.1. Replay attack:

Suppose an attacker intercepts a valid authentication request $[ID_{New} | ID_{tgs} | ID_{Auth} | TS_1 | TS_2 | Bloom\ Filter\ Index | KU_{New}]$ in the Authentication phase and tries to login to the Auth-Node by replaying it. By replying the captured authentication request, the attacker cannot gain access to network because attacker must know *PW* and *KR*. As a matter of fact, the TS_1 and TS_2 are embedded in the *BF* vector ($[ID_{New} | ID_{tgs} | Password | TS_1 | TS_2 | KU_{New}]$), and *BF* vector cannot be forged without knowing *PW*. Therefore, if Auth-Node detects duplicate TS_2, it will discard the replied request. In addition, without the knowledge of the *KR*, the attacker cannot perform ECDH, meaning that it will not have the shared key. Hence, after renewing the TS_1 by Auth-Node, the request will be rendered invalid. Thus, the proposed scheme can resist reply attack.

5.1.2. Sniffing:

If the attacker wants to sniff the communications, he cannot get much information because after phase 2, the messages are encrypted with the shared key and all the communications remain confidential. Also in phase 2, some information like *IDs* and *TSs* is exchanged clearly, but the important information like *PW* is embedded in the *BF* vector, which makes it impossible to trace back the *BF* vector to its inputs. Therefore, the sniffing attacks will not work on the proposed scheme.

5.1.3. Stolen Ticket/Verifier:

Suppose an attacker wants to intercept a Ticket and use it to gain access on the network. However, he cannot get the ticket because of its double encryption. The ticket is first encrypted with TGS-Node secret key ($\mathrm{En}_{tgs}(\text{Bloom Filter Index})$), and second with the user/node shared key ($\mathrm{En}_{Ks_A}[\mathrm{TS}_4|\mathrm{ID}_{tgs}|\mathrm{En}_{tgs}(\text{Bloom Filter Index})]$). The attacker has to know two secret key to access the network, which is not possible. On the other hand, if, by any chance, the attacker gets the original ticket, he will not be able to have session key by performing HMAC, for he does not have the information about the number of successful authentication. Therefore, he cannot have secret session key to communicate with Dest-Node. Thus, our scheme can overcome Stolen Ticket/Verifier attacks.

5.1.4. Denial-of-service attack:

DoS attack is a deadly attack on WSNs and the new scheme is well designed to prevent this attack. If the attacker wants to launch DoS on a user/node, the packet will be easily rejected because all users/nodes only communicate with authentic nodes e.g. TGS-Node, Auth-Node, Dest-Node which have previously been authenticated. Hence, all the communications are encrypted. To attack on Auth-Nodes, the attacker must have the knowledge of all users'/nodes' *ID* and *PW* to build *BF* vector. Even if the attacker creates a bogus message, the validation of *BF* vectors is computationally cheap. If the same *ID* fails more than three times in Authentication phase, the *ID* will be removed from valid *ID* list for a period of time. As explained above, the new scheme reduces the impact of DoS attacks.

5.1.5. Node Replication:

There are Node Replication attacks on WSNs like Sybil attack. However, in the new scheme after successful registration, the node cannot apply for another authentication. Additionally, each node

cannot use its ticket to access more than one node at a time. As explained before, to access Dest-Nodes, the ticket needs to be approved and verified by TGS-Node. Besides, if a node tries to communicate with more than one node at the same time, the TGS-Node will not approve that. Therefore, the new scheme overcomes these kinds of attacks.

5.1.6. Use of Tamper resistance:

In our scheme, we did not use Tamper resistance devices. These devices have extra cost on sensors. In addition, they are not compatible with all kinds of sensor nodes. They increase the energy usage of the nodes. To prevent attackers from getting nodes' information by capturing them, the important data are saved in hash mode on the sensors. In addition, the tickets have a limited lifetime and after expiring, the nodes should initiate the authentication phase again. Hence, if a node has previously misbehaved, the authentication operation will be denied.

The comparison results of our scheme with previous works are shown in Table 1. This table shows that the new scheme can resist more attacks than its predecessors. Because of these advantages, common attacks have much less effect on the network performance.

Table 1. Security analysis of different methods and proposed scheme

Attacks / Methods	Denial of Service	Replay Attack	Sniffing	Node Replication	Stolen Ticket/Verifier	Node capture	Use of Tamper resistance
$\varepsilon IBAS$	Y	N	Y	Y	Y	N	Y
Kumar	N	Y	Y	N	Y	Y	N
MAACE	Y	Y	Y	N/A	N/A	N/A	N
ENABLE	N	Y	Y	N/A	N/A	N/A	N
Al-Mahmud	N	Y	N	Y	Y	Y	N
Liu	N	Y	N	Y	Y	Y	N
IMBAS	Y	Y	N	N	Y	Y	N
HBQ	Y	Y	N	N	N	Y	N
Our Scheme	Y	Y	Y	Y	Y	Y	N
Y: means resistance against attack or yes in terms of using tamper resistance hardware, N/A: Not Available							

5.2. Simulation and performance evaluation

In this section, the performance of our proposed scheme regarding communication overhead and energy consumption on the MICA2 motes is evaluated as well as presenting a detailed analysis of our scheme compared to previous systems.

MICA2 mote works at 8 MHZ with an 8-bit processor ATmega-128, and adopts IEEE 802.15.4 standard. Table 2 shows the energy model employed for simulations in this study. The data in Table 2 were collected from previous works and our experiments [4, 11, 19, 22]. For Bloom Filter parameters, 7 Hash function with FNV and SHA-1 function were considered, with each index having 11 bit length. For symmetric key encryption, we implemented RC5 with 80-bit key size with security equal to 1024 RSA [32, 33]. For ECC, we used a key with 160 bits length. Each packet in IEEE 802.15.4 has 31 bytes as header, and total length of the packet is 128 bytes [4, 22]. For simplicity, we regarded the Auth-Nodes and TGS-Nodes as one entity called Main Nodes. For the Authentication phase and Access controlling phase, we presented the results in details as well as the results of all four main phases. For computing energy consumption, we used the equation (8) which determines the energy consumption for CPU and communication operations. For this equation, we had the value $V = 3v$ as power source. The value I and T for each operation are shown in Table 2.

$$E = V * I * T \tag{8}$$

Table 2. Energy model used for simulations

Operation	$I_{(mA)}$	$T_{(ms)}$	$V_{(v)}$
SHA-1 Hash function	8	3.636	3
Fowler–Noll–Vo hash function (FNV)	8	0.732	
MAC function	8	3.12	
Elliptic Curve Multiplication (P)	8	810	
Elliptic Curve exponential (Exp)	8	900	
Point Pairing (Pairing)	8	1900	
Symmetric encryption decryption (SYM)	8	0.26	
Transmitting 1Byte	27	0.645	
Receiving 1Byte	10	0.645	
Channel sensing	10	0.35	

Based on Table 2 and equation (8) we implemented an energy model class on NS2. For the simulation, we made use of NS2 open source software. The simulations ran in 500m*500m area with 500 node. Nodes were distributed in area randomly like usual WSNs. There were five clusters each of which had one TGS-Node and one Auth-Node. One base station was assigned to the area. In addition to our scheme, we implemented other works to compare their performance with each other. The results have been presented in Figure 8 and Figure 9.

Figure 8 shows the energy usage in Authentication phase. From computation overhead aspect, our scheme had less energy consumption in comparison to the others, for in this phase only 7 hashes were used, which were cheap operations on MICA2 (Fig. 8.). Fig. 9 shows the Average energy used by related parties for communication and computation in Authentication phase. According to the results, our scheme consumed less energy than the others. There are two main reasons for this outcome:

1) By using BF vectors and chaining PW to KU, there will be no need for public key certificates. Therefore, we reduced the communication overhead of transferring certificates and computation overhead of validating them.

2) We optimized the communication messages to reduce the overhead of transferring them. For example, we have forwarded BF indexes instead of forwarding the whole vector.

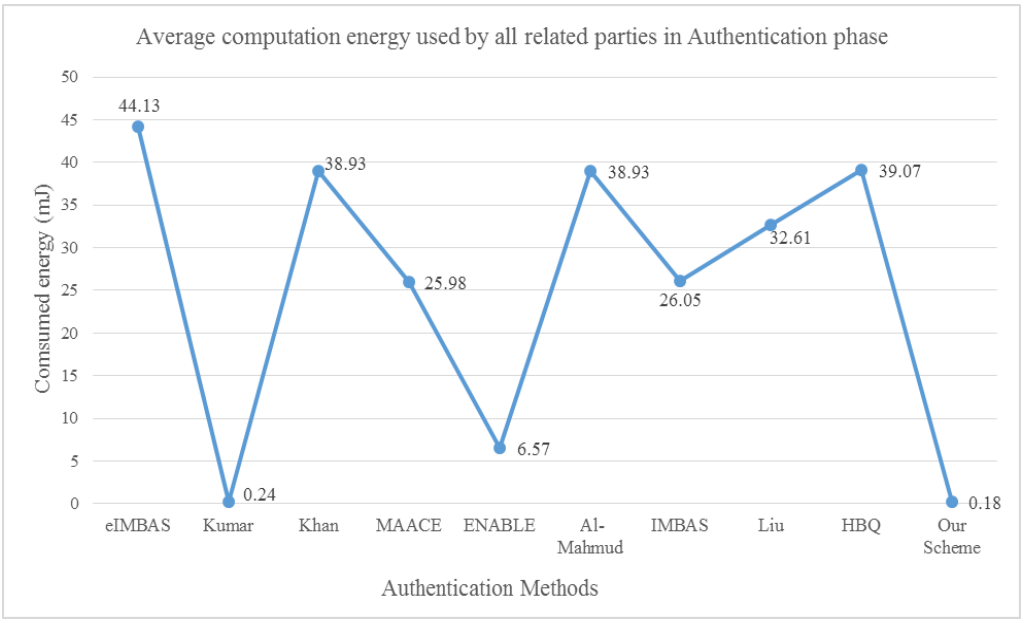

Fig. 8. Average computation energy used by all parties in Authentication phase,

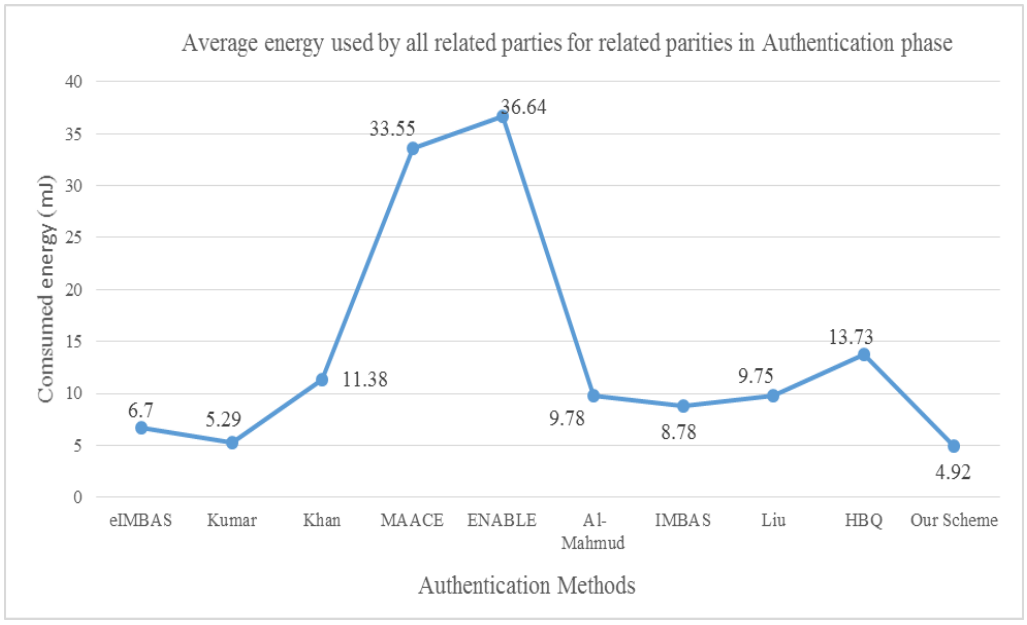

Fig. 9. Average energy used by related parties for communication and computation in Authentication phase[1]

The details of the authentication operation are shown in Table 3. The notation used for Table 3 is based on Table 2. We examined all protocols in detail, which may not be presented in the original papers. To clarify this, an example will follow. In MAACE scheme for main nodes, there is 3*P + 4*SHA + 1*SYM + 1*MAC. For authentication, the Main node[2] in this scheme verifies public key signature for message signature. Each signature verification needs one hash operation (1*SHA) and two elliptic curve multiplication (2*P). Additionally, the main node signs the results of authentication with its public key, requiring one hash operation (1*SHA) and one elliptic curve multiplication (1*P). In addition, it preforms one symmetric decryption (1*SYM), one MAC operation, and one hash. This detail is shown in Figure 8 of [11]. In their paper discussion, they did

[1] We have presented energy in mJ instead of µJ because numbers in µJ will be too large to be shown on figures

[2] They have used coordination node term for Main node term

not mention overhead of transferring signatures. Hence, they neglected the computational cost of verifying of signatures and certificates.

Table 3. Authentication phase's operations in detail

Node type / Methods	Main Nodes	User/Node	Dest-Node
$\varepsilon IBAS$	1*P + 1* Pairing+ 3*SHA	0	1*Exp + 1* Pairing+ 3* SHA
Kumar	2*SHA+1*SYM	6*SHA+1*SYM	0
Khan	0	3*P + 4*SHA	3*P + 4*SHA
MAACE	3*P + 4*SHA + 1*SYM + 1*MAC	3*P + 3*SHA + 1*SYM	2*MAC + 1*SYM
ENABLE	2*P+1*SYM+1*MAC	1*P + 1*SHA + 1*MAC	2*MAC + 1*SYM+ 1*SHA
Al-Mahmud	0	3*P + 3*SHA	3*P + 3*SHA
IMBAS	0	1*P + 1*SHA	3*P + 1*SHA
Liu	0	1*P + 1*SHA+ N*MAC	4*P + 2*SHA+ N*MAC
HBQ	0	1*P + 2*MAC	5*P + 2*MAC
Our	14*FNV	14*FNV	0

In the following are presented the Access Control phase simulation results in which we compared our scheme with ENABLE and MAACE since these were the only ones that support the Access Control services. ENABLE and MAACE use digital signatures for Access Control, which is an extremely expensive operation on MICA2, suffering from high-energy consumption. For example, the main nodes in these schemes perform one hash (1*SHA) and two elliptic curve multiplications. By comparison, our scheme uses only symmetric key methods and hash functions. For this reason, our scheme is much more energy efficient in this phase. Table 4 shows the total energy used in this phase by each entity.

Table 4. Total energy used in Access Control phase by each entity

Node type / Methods	Main Nodes	User/Node	Dest-Node
MAACE	21.01 mJ	0	48.48 mJ
ENABLE	21.01 mJ	0	48.48 mJ
Our	10.436 mJ	5.643 mJ	5.603 mJ

For more a more scrutinized comparison, we analyzed the total energy and time consumption of our method. Fig. 10. shows the average energy used by all nodes in complete Authentication and Access controlling process. The results indicated the fact that the new scheme has low energy consumption in total compared to the others, but Kumar's method is the lowest one in this chart. The reason of this outcome is that we utilized public key method for key sharing, which is more secure than other methods and scales well on WSNs compared to symmetric methods. In addition, Kumar's method does not support Access Control service. Moreover, compared to the other schemes employing public key methods, our scheme has lower cost and is more reasonable to use.

With regard to execution time, our scheme has the lowest run-time among the previous schemes. To achieve this result, we avoided complex operations and optimized our scheme to have less run-time (Fig. 11.). From the security perspective, run-time is a crucial parameter because prolonged run-time makes system prone to more attacks and exploits.

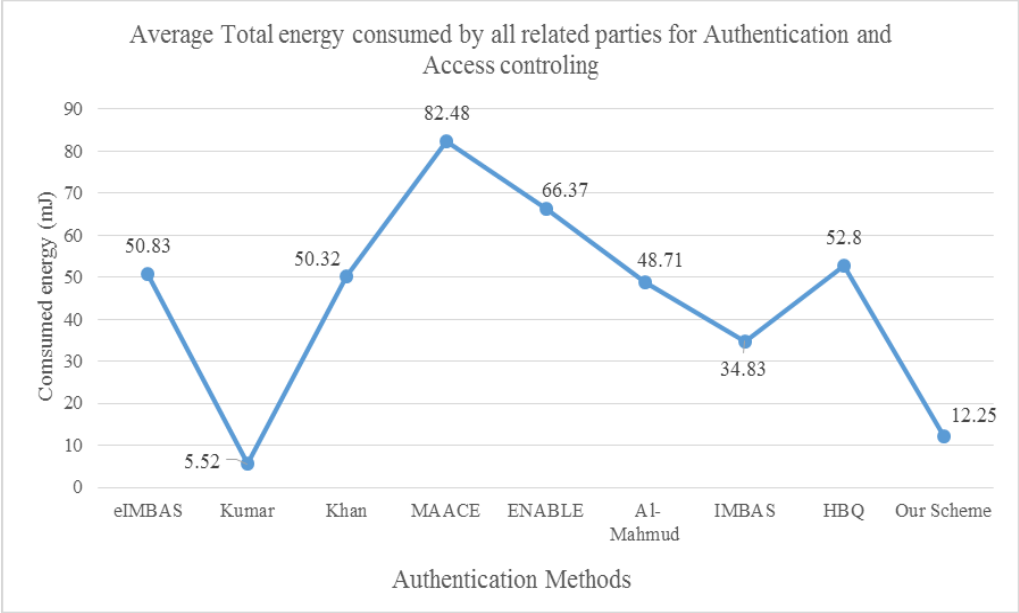

Fig. 10. Average total energy used by running through all phasesTotal run time of all phases.

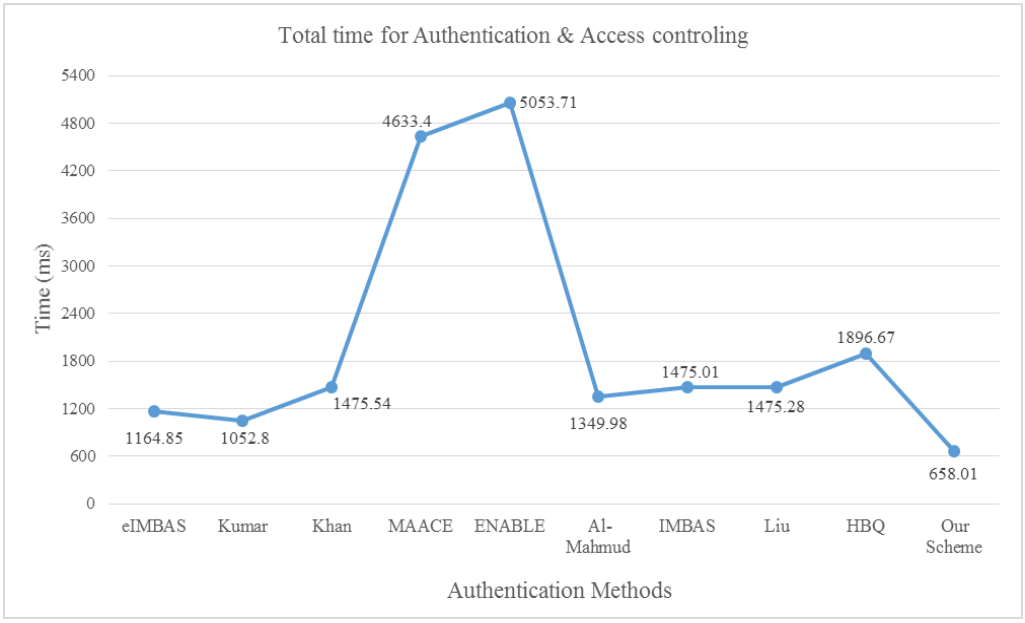

Fig. 11. Total run time of all phases

The above evaluations show that our scheme reduces energy consumption on the network by about 64.82% compared to the other public key schemes, which extends network lifetime. Additionally, it has an execution time of 658.01 ms, which is 37.49% faster than the lowest execution time among the previous schemes. In addition, our scheme offers more security services compared to the previous works, resulting a more robust framework.

6. Conclusion

In this paper, we have proposed a new Authentication and Access Control framework for large-scale hierarchical wireless sensor networks. The proposed scheme uses a hybrid approach of symmetric key and public key methods for confidentiality. Our scheme uses Bloom Filter in a novel way to eliminate the need for public key certificates. The simulation results show that our scheme is not only energy efficient, but it also has low execution time, which leads to more reliability against

attacks. We have reduced the Authentication and Access controlling time about 37.49%. The evaluations show that our scheme reduces energy consumption by about 64.82% compared to the other public key schemes, which extends network lifetime. In addition, the security analysis results indicate that the new approach provides various security services, and as result, it can withstand more attacks in comparison with previous works. We have considered some optional services like authentication forwarding that enables nodes to move freely across the network. For future works, researchers can use secure coding to change the TGS-Node and Auth-Node with other nodes depending on the situation.

References

[1] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: a survey," *Communications Surveys & Tutorials, IEEE,* vol. 11, pp. 52-73, 2009.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *Communications Magazine, IEEE,* vol. 40, pp. 102-114, 2002.

[3] W. Ben Jaballah, A. Meddeb, and H. Youssef, "An efficient source authentication scheme in wireless sensor networks," in *Computer Systems and Applications (AICCSA), 2010 IEEE/ACS International Conference on*, 2010, pp. 1-7.

[4] X. Cao, W. Kou, L. Dang, and B. Zhao, "IMBAS: Identity-based multi-user broadcast authentication in wireless sensor networks," *Computer Communications,* vol. 31, pp. 659-667, 2008.

[5] A. D. Dhawale, M. Chandak, and N. Thakur, "Authentication Techniques for Wireless Sensor Network," in *MPGI National Multi Conferences*, 2012, pp. 1-4.

[6] W. Ben Jaballah, M. Mosbah, H. Youssef, O. Ly, and A. Meddeb, "Modeling Source Authentication Protocols in Wireless Sensor Networks Using HLPSL," in *Network and Information Systems Security (SAR-SSI), 2011 Conference on*, 2011, pp. 1-9.

[7] W. Stallings, *Network Security Essentials: Applications and Standards, 4/e*: Pearson Education India, 2003.

[8] L. Tobarra, D. Cazorla, F. Cuartero, G. Diaz, and E. Cambronero, "Model checking wireless sensor network security protocols: Tinysec+ leap," in *Wireless Sensor and Actor Networks*, ed: Springer, 2007, pp. 95-106.

[9] Y. Liu, J. Li, and M. Guizani, "PKC Based Broadcast Authentication using Signature Amortization for WSNs," *Wireless Communications, IEEE Transactions on,* vol. 11, pp. 2106-2115, 2012.

[10] K. Sun, A. Liu, R. Xu, P. Ning, and D. Maughan, "Securing network access in wireless sensor networks," in *Proceedings of the second ACM conference on Wireless network security*, 2009, pp. 261-268.

[11] X. H. Le, M. Khalid, R. Sankar, and S. Lee, "An efficient mutual authentication and access control scheme for wireless sensor networks in healthcare," *Journal of Networks,* vol. 6, pp. 355-364, 2011.

[12] J. Zhang and V. Varadharajan, "Wireless sensor network key management survey and taxonomy," *Journal of Network and Computer Applications*, vol. 33, pp. 63-75, 2010.

[13] O. Delgado-Mohatar, A. Fúster-Sabater, and J. M. Sierra, "A light-weight authentication scheme for wireless sensor networks," *Ad Hoc Networks*, vol. 9, pp. 727-735, 2011.

[14] X. Fan and G. Gong, "Accelerating signature-based broadcast authentication for wireless sensor networks," *Ad Hoc Networks*, vol. 10, pp. 723-736, 2012.

[15] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. E. Culler, "SPINS: Security protocols for sensor networks," *Wireless networks*, vol. 8, pp. 521-534, 2002.

[16] P. Zeng, K. K. R. Choo, and D. Z. Sun, "On the security of an enhanced novel access control protocol for wireless sensor networks," *Consumer Electronics, IEEE Transactions on,* vol. 56, pp. 566-569, 2010.
[17] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM,* vol. 13, pp. 422-426, 1970.
[18] A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," *Internet Mathematics,* vol. 1, pp. 485-509, 2004.
[19] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," in *Cryptographic Hardware and Embedded Systems-CHES 2004*, ed: Springer, 2004, pp. 119-132.
[20] D. Hankerson, S. Vanstone, and A. J. Menezes, *Guide to elliptic curve cryptography*: Springer, 2004.
[21] E. Cayirci and C. Rong, *Security in wireless ad hoc and sensor networks*: John Wiley & Sons, 2008.
[22] K.-A. Shim, Y.-R. Lee, and C.-M. Park, "EIBAS: An Efficient Identity-based Broadcast Authentication Scheme in wireless sensor networks," *Ad Hoc Networks,* vol. 11, pp. 182-189, 2013.
[23] A. Al-Mahmud and R. Akhtar, "secure sensor node authentication in wireless sensor networks," *International Journal of Computer Applications,* vol. 46, pp. 10-17, 2012.
[24] A. K. Das, P. Sharma, S. Chatterjee, and J. K. Sing, "A dynamic password-based user authentication scheme for hierarchical wireless sensor networks," *Journal of Network and Computer Applications,* vol. 35, pp. 1646-1656, 2012.
[25] H. Wang, B. Sheng, and Q. Li, "Elliptic curve cryptography-based access control in sensor networks," *International Journal of Security and Networks,* vol. 1, pp. 127-137, 2006.
[26] X. H. Le, S. Lee, I. Butun, M. Khalid, and R. Sankar, "An Energy-Efficient Access Control Scheme for Wireless Sensor Networks based on Elliptic Curve Cryptography," *Journal of Communications and Networks,* vol. 11, p. 599, 2009.
[27] B. Nouri-Moghaddam and H. R. Naji, "Improving HBQ Authentication and Access control in wireless sensor network," in *Information and Knowledge Technology (IKT), 2013 5th Conference on*, 2013, pp. 82-87.
[28] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation,* vol. 48, pp. 203-209, 1987.
[29] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology—CRYPTO'85 Proceedings*, 1986, pp. 417-426.
[30] H. Wang and Q. Li, "Efficient implementation of public key cryptosystems on mote sensors (short paper)," in *Information and communications security*, ed: Springer, 2006, pp. 519-528.
[31] W. Diffie and M. E. Hellman, "New directions in cryptography," *Information Theory, IEEE Transactions on,* vol. 22, pp. 644-654, 1976.
[32] "http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/key-size.htm," 2015.
[33] "http://www.nsa.gov/business/programs/elliptic_curve.shtml," 2015.