CHAPTER 4

ENERGY EFFICIENT AUTHENTICATION USING DIGITAL SIGNATURE ALGORITHM

Message authentication is a mechanism to verify that received messages come from the alleged source and have not been altered. It also verifies sequencing and timeliness. Digital signature is an authentication mechanism that also includes measures to counter repudiation by the source.

For providing authentication public key approaches are suitable, but due its slow nature and power hungry nature it is not preferred for light weighted sensor devices, so symmetric key approaches are preferred compare to asymmetric one. But symmetric approaches do not give complete security services and one way authentication. Here we have applied public key approaches in such a way that total energy consumption decreases as compare to traditional one, as well as we are able to gain full security services.

Our Thesis is concerned for those application areas where authentication is the major concern especially in battle-field scenarios, where fraudulent node impersonates as a legitimate sensor. So in this chapter, we are presetting Energy efficient authentication scheme using DSA [8] by applying offline signature scheme [19].

Using the preprocessing/offline approach, in this chapter we have presented an Energy efficient authentication scheme using DSA. In our thesis, we have presented three different authentication schemes as follows, so one can use
these authentication schemes as multilevel authentication, where authentication is the major concern than energy. The multilevel authentication consist of following three stages authentication

- The first level authentication is achieved through the public key distribution using EEDHA/ EEECC as discussed in the previous chapter.
- In this chapter the second level authentication is achieved by proposed digital signature standard in energy efficient manner referred as EEDHA. We have identified and shifted those steps of DSA from node to base station, which are not necessarily to be performed on sensor node. Some of the signature component does not depend upon the message at all and can be preprocessed earlier before the deployment.
- The third level of authentication is achieved and discussed in next chapter. This level of authentication can be found by choosing such encryption algorithm having initialization vector or a chosen counter value that provide authentication with energy efficient confidentiality among the nodes.

4.1 Authentication

Authentication in sensor networks is a complicated task. In traditional networks, that have not the constraints of sensor networks, the common way to authenticate is using public key cryptography (PKC). The sending entity signs the message using its private key and the receiving entity verifies the authenticity of the entity/message with the public key of the sender. In sensor networks, it is typically assumed that public key cryptography cannot be used, because of the resource constraints. In that case authentication must be provided using symmetric cryptography.
4.2 Cryptographic Primitives for Authentication

Any message authentication or digital signature mechanism has two levels of functionality. At the lower level, there must be some sort of function that produces an authenticator: a value to be used to authenticate a message. This lower-level function is then used as a primitive in a higher-level authentication protocol that enables a receiver to verify the authenticity of a message.

Entity and message authentication is provided by public-key schemes as digital signature and by symmetric-key scheme as message authentication code (MAC). Following two cryptographic primitives typically used for authentication in WSNs are:

- Message Authentication Code
- Digital Signature

4.2.1 Message Authentication Code

A Message Authentication Code (MAC) [7] is based on symmetric key cryptography to provide message authentication. To protect a message via a MAC, the sender and the verifier(s) first need to share a secret key, identified as MAC key. The sender then generates the MAC value of Message by passing it through an algorithm, like as hash function, called MAC algorithm.

A MAC algorithm takes inputs consisting a message M and a MAC key K, and outputs MAC value, i.e., MAC is $\text{MAC}_K (M)$. The message M jointly with the MAC value is sent to the verifier(s). A verifier uses the same MAC algorithm to authenticate a message and the shared MAC key
and generates a MAC value of the received message. The generated MAC value and the received MAC value are then compared against each other to authenticate the message. Following Figure shows the working of MAC.

![Figure 4.1: Operation of Message Authentication Code](image)

The security of a MAC lies in the secret shared MAC key K. Given M and MAC, it is hard to find K. Without MAC key, no one cannot generate a correct MAC, and the message looks authentic. A MAC is efficient and fast mechanism to compute on sensor nodes and reduces the memory consumption. Though, it cannot provide source authentication in broadcast scenario, where the same MAC key is shared amongst more than one sender and one can verifier to generate and verify MAC values.

### 4.2.2 Digital Signature

A digital signature [13] is based on public key cryptography (PKC) to ensure authentication. Digital signatures provides a method to assure that a
message originates from the person, who claims to have generated the message (source authentication) and also ensures that the contents of the message have not been altered in transit. In contrast to a single shared MAC key, each user now owns a matched pair of private and public keys. The private key is kept secret by the user, whereas the public key is made available to everyone as shown.

![Figure 4.2: Operation of Digital Signature](image)

The security of a digital signature deceit in the secrecy of the private key. The intuitive security notion for digital signatures is the impossibility to forge a user's signature without knowledge of his private key. Figure 4.2 illustrates the use of digital signatures.

- **Properties of digital signatures**

  Message authentication protects two communicating parties, who exchange messages from any third party. Though, it does not protect the two parties against each other. Numerous forms of dispute between the two are possible.
Where there is not complete trust between sender and receiver, something more than authentication is required. The most striking solution to this problem is the digital signature. The digital signature must have the following properties:

- It should verify the author, the date and time of the signature.
- It should authenticate the contents at the time of the signature.
- It should be verifiable by third parties, to resolve disputes.

**Requirements of Digital Signature**

On the basis of the above properties [13], we can formulate the following requirements for a digital signature.

- The signature’s bit pattern should depend on the message being signed.
- The signature must use some other information unique to the sender to prevent both forgery and denial.
- It must be easy to produce the signature.
- It must be easy to recognize and verify the signature.
- It must be computationally infeasible to generate a digital signature, either by constructing a new message for an existing digital signature or by making fraudulent digital signature for a given message.
- It must be practical to keep a copy of the digital signature in storage. A secure hash function provides a basis for satisfying these requirements.
4.3 Literature Survey

Authentication is the proving of one's identity. Authentication in WSNs is a challenging task due to the nature of the resource constraint network. This section reviews the existing solutions to authentication challenges in WSNs, focused on two authentication problems, broadcast authentication and outside/offline user authentication.

In conventional networks, authentication is achieved via digital signatures, i.e., via PKC. Though, PKC had been considered too resource hungry to use in WSNs because of its large processing computation and storage requirements. Thus, the security solutions for the WSNs were considered only by using symmetric cryptography for a long period of time. [9] showed the possibility of both ECC and RSA on 8-bit processors with ECC performance advantage above RSA. Because PKC has now also become a part of WSN security schemes. The following sections review the previously proposed schemes of broadcast authentication and outside/offline user authentication and their lack to address certain issues.

4.3.1 Broadcast Authentication

Existing proposed work in the area of broadcast authentication in WSNs can be categorized as symmetric cryptographic schemes and asymmetric cryptographic schemes (digital signature based schemes).

4.3.1.1 Proposals Based on Symmetric Cryptography

In WSNs, authentication was generally provided through Message Authentication Code (MAC), a symmetric cryptographic approach. In a
pair-wise MAC based authentication scheme, the MAC generated by a sender is verified by the receiver using the same MAC key, already shared between the two entities. This kind of approach provides an efficient solution to the pair-wise authentication problems in WSNs.

In a broadcast condition, the MAC key used to authenticate the broadcast messages wants to be shared among all potential receivers. Furthermore the compromised node can send false messages on behalf of a legal sender. Therefore, a typical MAC based authentication scheme can only provide mutual (pair-wise) authentication in WSNs and cannot handle the broadcast/one way authentication.

- **µTESLA**: Solution of above problem was given by µTESLA [2], is the symmetric cryptographic scheme to deal with the problem of broadcast authentication in WSNs. The major symmetric cryptography broadcast authentication schemes are different variations of µTESLA scheme for WSNs. These schemes can also be further categorized depending on the type of broadcast senders.

  Broadcast authentication technique achieves asymmetric properties and therefore enables low powered nodes to perform source authentication. µTELSA uses shared symmetric key between the base station and each of the nodes. This key work as a seed value for production of MAC. µTELSA has the drawback of requiring a shared symmetric key between a base station and every sensor nodes.

- **SNEP**: In SNEP each node share a unique master key with the base station. This master key is used to obtain all other keys. For data
encryption SENP employs a onetime key produced by using a key and an incremental counter as input to the RC5 cryptographic algorithm. The RC5 algorithm outputs string that is used as the one time key. The message is XORed with the Onetime Key and transmitted and the counter is incremented to send the next message. The base station, alert of the node’s counter value and the derived key, produce the identical one time key, XORs the encrypted message with the onetime key to produce the clear text.

The first drawback of this method is the distribution of the master key and management of the key at base station. Secondly if the master key is compromised there will be no security.

4.3.1.2 Schemes Based on Asymmetric Cryptography

Numerous digital signature scheme based broadcast authentication schemes for WSNs have been proposed in [5][6][18] addressing numerous limitations of µTESLA based schemes. All these schemes presume resourceful devices as the broadcast senders and not typical sensor nodes. The schemes discussed in [5] use the conventional PKC based signature schemes, where a sender signs a message using his private key and the receiver verifies the signed message using the sender's public key. Really they provide a solution to the public keys and certificates management problem faced in WSNs. A number of other schemes [5] use ID-based signature schemes.
[5][6] Discusses a certificate based authentication scheme for WSNs. Certificate contains the public key of sender signed by the private key of the base station. A broadcast sender signs a message using its private key and broadcasts the signed message with its certificate. On receiving a message, the receiver first verifies the signed certificate by using the public key of the base station.

- **ID-based Signature:** A novel type of public key cryptography named as Identity-based cryptography or ID-based cryptography [1] to replace the usual certificate based public key cryptography(PCK) has been introduced. Implementing the traditional public key cryptography based signature schemes has the following requirements which do not outfit WSNs environment:

  1. Managing the public key infrastructure (PKI) is difficult (particularly in a WSN environment).

  2. Storing the Public keys on each receiver to verify signed messages (increased storage overhead).

  3. To stay away from storage overhead, signed certificates can be sent along the signed message to obtain public keys (that increased transmission overhead).

  4. Receivers should validate the signed certificate earlier than using public key (increased computation overhead).

    In difference to certificate based public key cryptography, ID-based cryptography replaces a user's public key by his unique public
identifier (ID), such as email address, physical IP address etc., that uniquely identifies him. The similar private key is generated by a trusted third party, a private key generator (PKG). The PKG generates a master secret key $m_{sk}$ and a master public key $m_{pk}$ (called Setup phase).

- **TinyPK**: TinyPK uses the small exponent variant of the well known RSA public key cryptosystem to implement authentication of outside party, Diffie-Hellman key agreement and authentication of sensor network to external party. An outside party is a unit that wishes to establish secure communication with the sensor network [15].

  The private part of RSA is not at all carried out on the nodes, this is done at the certificate authority (CA). The nodes only want to implement the public arts, i.e. data encryption and signature verification, which are much faster than the private parts in RSA. There are not any certificates used so there is no real-time interaction with the CA, which means that the CA’s public key must be pre-loaded in every node [15].

- **Rabin's Scheme**: It was introduced in [10]. It is based on the factorization problem of large numbers and therefore similar to the security of RSA with the similar sized modulus. Rabin's Scheme takes asymmetric computational cost. The encryption operation is tremendously fast, however decryption times are comparable to RSA.
Here is a brief description of the Rabin's Scheme. More detailed description and the mathematical proofs is available in [10].

### 4.3.2 Limitation of existing schemes

All the symmetric and asymmetric cryptography based authentication schemes, presented in the literature suffer from significant limitations. In case of symmetric schemes (μTESLA based schemes), these limitations are the broadcast at regular and predefined intervals, the storage of μTESLA parameters, the distribution of key chain commitments and the delayed authentication. μTESLA based schemes fail to provide a solution to the real-time applications of WSNs. In case of asymmetric schemes (digital signature based schemes), these limitations are the management of public keys and certificates and the cost of applying PKC on sensor nodes particularly the time cost. The previous raises the scalability problem and the latter is critical for real-time applications. Also, all of the schemes apart from [16][4] assume broadcast senders to be powerful devices and not ordinary sensor nodes.

### 4.3.3 Distributed Schemes

It is proposed in [20] which first addressed the outside/offline user authentication problem in WSNs. This scheme realized the presence of compromised sensor nodes in the network affecting the user authentication process and introduced notion of a threshold authentication.
PKC based distributed user authentication scheme discussed in [21], says that every user obtains a private key from the base station and a certificate containing the user's corresponding public key. This certificate is signed by the base station’s private key of the base station.

In this scheme, the receiving sensor nodes verify two signatures for every authentication request; one to verify the signed user's certificate and second to verify the signed nonce. Therefore, verifying a user is expensive for a sensor node in terms of computation cost. This method is prone to DoS attack, against the sensor nodes storage. In addition, the periodic broadcast of the public key by the base station to all sensor nodes increases the communication overhead.

This approach only works well, where the sensor nodes are deployed in a grid form. Furthermore, it results in increased storage overhead, because each used token is stored on more than one sensor node in the network. Due to the enlarged storage overhead, it restricts the number of outside users of WSNs [14]. This method also results in increased communication overhead because for every user request more than one sensor node are consulted to detect token re-usability.

4.3.4 Digital Signature Schemes

A digital signature is a mathematical system for demonstrating the authenticity of a digital message or document. Valid digital signature makes a recipient to believe that the message was created by a known sender, and that it was not changed in transit. The various digital signature schemes are RSA digital signature scheme [13], ElGamal
digital signature scheme [3], Schnorr digital signature scheme [13], Digital Signature Standard (DSS) scheme [8] and elliptic curve digital signature scheme have been presented.

- **Elgamal digital signature scheme**: Before examining the Digital Signature standard proposed by NIST, it will be supportive to understand the ElGamal and Schnorr signature schemes [17]. The ElGamal encryption scheme is designed to allow encryption by a user’s public key with decryption by its private key. The ElGamal signature scheme uses the private key for encryption and the public key for decryption.

  The ElGamal signature scheme is a digital signature scheme which is based on the difficulty of computing discrete logarithms. The ElGamal signature scheme allows a third-party to confirm the authenticity of a message sent over insecure channel. A third party can forge signatures either by finding the signer's secret key $x$ or by finding collisions in the hash function $H(m) \equiv H(M) \mod (p - 1)$.

  The signer must take cautious to choose a different $k$ uniformly at random for each of the signature and certain that $k$, or even partial information about $k$, is not disclose. Or else, an attacker may be able to deduce the secret key $x$ easily, perhaps enough to permit a practical attack. If two messages are sent using the same value of $k$ and the same key, then an attacker can compute $x$ straightly.

- **Schnorr signature**: The Schnorr signature scheme [11], [17], derived from the Schnorr identification method, through the Fiat-Shamir transform,
is one of the earliest discrete log-based signature schemes proposed in the text [17]. Its simplicity and efficiency has attracted considerable attention.

4.4 Comparison between RSA and DSA

The DSA is designed to provide only the digital signature and data integrity function. Unlike RSA, it cannot be used for encryption or key exchange, but, it is a public-key technique. Figure 4.3 [12] shows the DSA approach for generating digital signatures to that used with RSA.

In the RSA approach, the message to be signed is input to a hash function that produces a secure hash code of fixed length. Then this hash code is encrypted using the sender’s private key forming the signature. The message and the signature both are then transmitted to receiver. The recipient takes the message and produces a new hash code. The recipient also decrypts the signature using the public key of the sender. If the calculated hash code matches the coming decrypted signature, the signature is accepted as valid. Since only the sender knows the private key, so the sender could have produced a valid signature only.

The hash code used in DSS provides as input to a signature function along with a random number generated for this particular signature. Also the signature function depends on the sender’s private key and a set of parameters identified to a group of communicating principals. We can view as this set to constitute a global public key. The result is a signature consisting of two components, labeled s and r [12].
At the receiving end, the hash code of the incoming message is then generated. Addition with the signature is input to a verification function, that is also depends on the global public key as well as the sender’s public key, which is paired with the sender’s private key. Output of the verification function is a value that is equal to the signature component if the signature is valid. The signature function is like that only the sender, with knowledge of the private key, can produce the valid signature.

In terms of signature size and number of bits required to generate the same security level, DSA is better than RSA. Following figure shows how the size of signature increases with varying key size for DSA and RSA [3].

**Figure 4.3:** RSA and DSA approach of digital signature
Figure 4.4: Comparison of RSA and DSA signature sizes

Also the DSA key size is smaller than the key size of RSA for producing the same security level. Following table shows the key sizes for providing same security.

Table 4.1: Key size comparison between DSA & RSA

<table>
<thead>
<tr>
<th>RSA Key size</th>
<th>DSA key size</th>
</tr>
</thead>
<tbody>
<tr>
<td>512</td>
<td>113</td>
</tr>
<tr>
<td>704</td>
<td>129</td>
</tr>
<tr>
<td>1024</td>
<td>153</td>
</tr>
<tr>
<td>1536</td>
<td>183</td>
</tr>
<tr>
<td>2240</td>
<td>218</td>
</tr>
</tbody>
</table>

4.5 Problem Statement

There are several approaches available in literature for providing the authentication in sensor networks using symmetric and asymmetric key approaches [2]. Symmetric approaches does not provide one-way authentication, i.e; only available in public key cryptography through digital
signature schemes like RSA, DSA etc. The existing distributed user authentication schemes for providing digital signature results in high processing cost [20] [21] and storage overhead [14]. So again public key approaches are costly than symmetric key approaches and not suitable for light-weighted sensor nodes.

4.6 Proposed solution

In this chapter, by using the preprocessing/offline approach [19], we have presented an Energy efficient authentication scheme using DSA, because of requirement of a lightweight user authentication mechanism. So here we have presented lighted weighted distributed solution for authentication in sensor network.

Indeed, a user could pre-calculate a number of values to be used to sign documents as needed known as offline signature [19]. The only other somewhat demanding task is the determination of a multiplicative inverse. Again, a number of these values can be pre-calculated at base station. It will save a significant computation time and energy at sensor node.

To produce the authentication, DSA have two components for generating the signature (r and s). The first component of the signature and some other components that do not depend on the message can be preprocessed at base station. By this we can save a significant amount of energy at node. Another advantage of using the DSA is that it possesses discrete logarithm property [12].
So we have proposed energy efficient authentication using DSA [18] [8]. Here we have applied DSA in such a way that total energy consumption of the sensor node decreases.

We have carried out the experiment at different number of sensor nodes. The result shows time saved by sensor network at random deployment and random selection of source node.

4.7 Digital signature standard

DSA is based on the difficulty of computing discrete logarithms [12] as discussed below.

4.7.1 Discrete logarithm

Digital signature scheme have the properly of desecrate logarithm. It ensures that the intruder cannot calculate the private key knowing the public key of the legitimate node In mathematics, specifically in abstract algebra and its applications, discrete logarithms are group theoretic analogues of ordinary logarithms. In particular, an ordinary logarithm $log_a (b)$ is a solution of the equation $a^x = b$ over the real or complex numbers. Similarly, if g and h are elements of a cyclic group G then a solution x of the equation $g^x = h$ is called a discrete logarithm to the base g of h in the group G[12] [3].

Discrete logarithms are perhaps simplest to understand in the group $Z_p^x$. This is the set $\{1, ..., p-1\}$ of congruence classes under multiplication modulo the prime p. If we want to find the $k^{th}$ power of one of the numbers, we can do so by finding its $k^{th}$ power as an integer and then
finding the remainder after division by p. This process is called discrete exponentiation. Discrete logarithm is just the inverse operation.

### 4.7.2 DSA Approach and its Analysis

It is based on schemes originally presented by ElGamal [3] and Schnorr [11].

- **Global Public element selection**

  A prime number q is chosen. Next, a prime number p is selected such that (p-1) divides q. Finally, g is chosen to be of the form $h^{(p-1)/q} \mod p$, where h is an integer between 1 and (p-1) with the restriction that must be greater than 1.

- **Selection of private and generation public key**

  Each user selects a private key and generates a public key. The private key must be a number from 1 to (q-1) and should be chosen randomly or pseudo randomly. The public key is calculated from the private key as

  $$y = g^x \mod p$$

- **Signing**

  To create a signature, a user calculates two quantities r and s as follows

  $$r = (g^k \mod p) \mod q$$

  $$s = [k^{-1}(H(M) + xr)] \mod q$$
• Verification

At the receiving end, verification is performed

\[ w = (s')^{-1} \mod q \]

\[ u1 = [H(M')w] \mod q \]

\[ u2 = (r')w \mod q \]

\[ v = [(g^{ul}y^{u2}) \mod p] \mod q \]

If the value of \( v \) matches the component of the signature, then the signature is validated, i.e. message is accepted if and only if \( V=r' \)

4.7.3 Analysis of Digital signature algorithm

From the structure of the algorithm, as discussed above, following are the quite interesting points

1. Note that the test at the end is on the value, which does not depend on the message at all. Instead \( r \) is a function of \( k \) and the three global public-key components. The multiplicative inverse of \( k \) is passed to a function that also has as inputs the message hash code and the user’s private key. The structure of this function is such that the receiver can recover \( r \) using the incoming message and signature, the public key of the user, and the global public key.

2. Given the difficulty of taking discrete logarithms [12], it is infeasible for an opponent to recover \( k \) from \( r \) or to recover \( x \) from \( s \).
3. Another point worth noting is that the only computationally demanding task in signature generation is the exponential calculation. Because this value does not depend on the message to be signed, it can be computed ahead of time.

4. Indeed, a user could pre-calculate a number of values of $r$ to be used to sign documents as needed. The only other somewhat demanding task is the determination of a multiplicative inverse $r^{-1}$. Again, a number of these values can be pre-calculated.

Above properties of pre-calculation and algorithm independency on the message motivate us to implement this standard algorithm in sensor network. In next proposed section, we have performed the above said parameters at base station and saved a lot of sensor node energy.

### 4.8 Proposed scheme: Energy efficient Authentication using Digital signature Algorithm (EEDSA)

To provide the authentication in sensor network, following is the proposed DSA in energy efficient manner for sensor network. Our method is elaborated in three phases as follows.

#### 4.8.1 Phase-1- Before deployment of the sensors/offline Processing

We choose three parameters at base station that are public and can be common to a group of sensor. Selection of Global Public-Key Components is already discussed in the above section.

To choose these parameter a prime number is chosen. Next, a prime number $p$ is selected with such that $q$ divides $(p - 1)$. Finally, $g$ is chosen to
be of the form $h^{(p-1)/q} \mod p$, where $h$ is an integer between 1 and $(p-1)$ and with the restriction that must be greater than 1 as follows.

$p$ prime number where $2^{L-1} < p < 2^L$

$q$ Prime divisor of $(p-1)$,

$g \equiv h^{p-1} \mod p$, where $h$ is any integer with $1 < h < (p-1)$ such that $h^{(p-1)/q} \mod p > 1$

- **Selection of private keys & generation of public keys:** Next we choose different private keys for each sensor and calculate its corresponding public key and deploy pair of private & public to each sensor as follows.
  - **Sensor’s Private Key**
    
    $x$ random or pseudorandom integer with $0 < x < q$.

  - **Sensor’s Public Key**
    
    $y = g^x \mod p$

- **Creation of partial signature at the base station**

  To create signature, user calculates two quantities, $r$ and $s$, that are functions of the public key components $(p, q, g)$, the user’s private key ($x$), and the hash code of the message $H(M)$ and an additional integer $k$ that should be generated randomly or pseudo randomly and be unique for each signing as follows.
Because in signing r is not depended on message, so we pre calculate the value of r and value of $K^{-1}$ and deploy it to the sensors, before the deployment of the sensors with their public key, where r is calculated as

$$r = (g^k \mod p) \mod q,$$

The Signature is given as components of $(r, s)$, Where k is random or pseudorandom integer with $0 < k < q$. Deploy the signature value at the corresponding sensors with their public key.

### 4.8.2 Phase-II- After deployment of the sensors/ Online processing

After loading the signing function and public key, we deploy the sensors in random fashion and after that the next bootstrapping phase starts.

- After deployment each sensor shares its public key to its neighbors.
- After sensing the message and calculating the message hash $H(M)$, the sensor calculates rest of signature part $s$ as follows.

$$s = [k^{-1}(H(M) + xr) \mod q$$

- The sensor forwards this message to its next hop with its signature.

The receiving sensor/ base station verify the coming message by its signature as follows, and if it find legitimate then it forward/accept this message with its own signature to next hop.

$$w = (s')^{-1} \mod q$$

$$u1 = [H(M')w] \mod q$$

$$u2 = (r')w \mod q$$
\[ v = [(g^{ul} y^{u^2}) mod p] mod q \]

Where \( v \) is the function of the public key components, the sender’s public key, and the hash code of the incoming message.

If \( v = r \), message is accepted and forward it to next hop else reject the coming message. The following diagram shows the working of above phase.

**Figure 4.5:** Proposed model of DSA for Sensor network
4.9 Result and Analysis

From the above figure, we can see that we save those energy steps of the DSA at sensors nodes, which are involved for the calculation of \( r \) & \( k^{-1} \) by shifting these to the base station.

Calculation of \( r \) & \( K^{-1} \) at base station before deployment saves the CPU computation time as follows.

Implementation Assumptions

- Using 8051 microcontroller in sensor of XLAT=11.0592 MHz.
- Continuous power supply i.e. resistance, capacitance & inductance are static 12 clock per machine cycle
- Let private key & value of \( k \) is 255.
- Time taken for calculation of \( r =(12/11.0592) \times \text{number of machine cycle} \)
- Time taken for calculation of \( k^{-1} = (12/11.0592) \times \text{number of machine cycle} \)
- Total time saved \((T_{\text{CPU,Computation}}) = \text{Time taken for calculation of } r + \text{Time taken for calculation of } k^{-1}\).
- Total Energy saved \((E) \approx K1 \times T_{rfm} + K2 \times T_{\text{CPU,Computation}} \) where \( T_{\text{CPU,Computation}} = \text{time taken by the CPU at increased bit rate} \), \( T_{rfm} = \text{time taken by the transmission of data} \). K1 and K2 are the constant that depend on the current consumption of the RFM and CPU respectively at chosen frequency and transmission power. Then Total time saved by the CPU is...
\[ T_{\text{CPU SAVED TIME}} = 3.597 + 460.0694 = 463.66614 \text{ } \mu s/ \text{ per sensor} \]

\[ = 0.4636661 \text{ ms/sensor.} \]

**Figure 4.6:** Saved time at sensor node for different sensors

The above figure shows the total time saved by our proposed scheme using EEDHA in the sensor networks. The experiment is carried at different density of up to 1000 sensor node in a network. Also three different experiments are carried out at different RF sensors of networks. As we can see that, if we are increasing the number of sensors in the network the processor consumption time for computation will be saved more in proposed scheme.

### 4.10 Conclusion of the Chapter

We have presented a distributed light weighted authentication protocol for sensor network using public key cryptography. Digital signature algorithm scheme for authentication is proposed in energy efficient manner, such that total computation times at sensor node decreases and consequently decreases...
the energy consumption due to the proportionality with time, and increases its life time. This has been done by identify and preprocessed some of the steps of proposed scheme to base station in offline manner.

We are also getting authentication by two more different types of mechanism beside this in energy efficient way. The first level authentication achieved through the key distribution using EEDHA & EEECC and the next level authentication achieved by using an initialization vector and chosen counter as authenticator used in encryption discussed in next chapter. So we can go for three different level of authentication for sensor network of those application areas where authentication is having bigger concern than energy.

4.11 Summary of the chapter

This chapter presents energy efficient authentication using digital signature algorithm in sensor networks. The Initial part of the chapter contains authentication services along with properties and requirements of the digital signature.

The next section of literature survey contains the authentication mechanism available in sensor networks based on broadcast and asymmetric approaches. Based on the pros & cons of existing approaches, problem is identified in the subsequent section. The proposed solution is given with their result and analysis using energy efficient digital signature algorithm at the last of the chapter.
REFERENCES


[17] Wei Zhao ; Yong Peng ; Feng Xie, ; Dongqing Chen (2012), 'Nominative signature scheme based on Schnorr signature', Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on Volume: 1 Digital Object Identifier: 10.1109/CSAE.2012.6272694, Page(s): 724 - 728


