International Journal of Latest Trends in Engineering and Technology
IJLTET Special Issue- ICRACSC-2016 , pp.125-135
e-ISSN: 2278-621X

ENERGY EFFICIENT SECURE KEY MANAGEMENT SCHEMES FOR WSNS AND IOT

Dr. M.Suresh Babu[1], Md.Raziuddin[2] and Dr.D.Asha Devi[3]

Abstract: Secret sharing is critical to most applications making use of security and remains one of the most challenging research areas in modern cryptography. In this paper, we propose a novel efficient multi-secret sharing scheme based on the Chinese remainder theorem (CRT) with two verification methods, while the previous works are mostly based on the Lagrange polynomial. Key management schemes play an important role in communication security in Wireless Sensor Networks (WSNs). While the previous works mainly targeting on two different types of WSNs: distributed and hieratical, in this paper, we propose our flexible WSN key management scheme, which is based on (*n,t,n*) multi-secret sharing technique, to provide a key management solution for heterogeneous architecture. The powerful key managers are responsible for most of the communicational and computational workload. Internet of Things (IoT) becomes more and more popular and practical in recent years. Considering the diversity of the devices and the application scenarios, it is extremely hard to couple two devices or sub-networks with different communication and computation resources. In thispaper, we propose novel key agreement schemes based on (*n,t,n*) multi-secret sharing techniques for IoT in order to achieve light weighted key exchange while using Host Identity Protocol (HIP). We refer the new schemes as HIP-MEXs with different underlying multi-secret sharing techniques.

Keywords : Chinese Reminder Theorem (CRT), Host Identity Protocol (HIP),Wireless Sensor Networks.

I. INTRODUCTION

In recent years, security and privacy become more and more important in our daily life. Sensitive data are kept in various devices such as cell phones and personal computers; however, most of them are not well protected or under poor supervision while malicious attacks who are targeting sensitive information are growing extremely fast. The awareness of the importance of the security and privacy makes the topic of security very popular. There are a lot of ways to achieve security and privacy, and among all those methods, cryptography is the most important and fundamental one. Cryptography has a very long history. Cryptographers and mathematicians worked for centuries to find better ways to encrypt the plaintexts as well as try to decrypt the ciphertexts. It was until 1976 [1], the society of cryptography ushered the biggest or the only "revolution" brought by Diffie and Hellman's public key cryptography. However, the advanced encryption technologies are still "not safe enough". On the contrary, because of the progress people have made in computer hardware and software, the electronic devices become more and more powerful which makes it much easier for a malicious attacker to crack the poorly encrypted information. Wireless Sensor Networks (WSNs) are widely accommodated in different areas performing various functions.

1.1 Overview of Wireless Sensor Networks : Wireless sensor networks (WSNs) become more and more feasible and widely used since Micro-Electro-Mechanical system (MEMS) was introduced to manufacture small sensor devices [2]. A sensor node is a resource constrained device with one or more sensors integrated. Those sensors may have different functions, such as detecting and collecting different environment variables. And because of the sensor nodes are typically deployed in locations which are difficult to access, the collected data needs to be transmitted and reported to a base station or a sink through wireless communication techniques. Thousands of sensor nodes are usually deployed within a large area in order to monitoring and reporting the collected physical parameters [3]. Because of the limited energy resource that a sensor node has, its communication range tends to be short, thus making almost impossible for two nodes located far apart to communicate directly with each other.

[1] *Department of CSE, NallaMalla Reddy Engineering College, Hyderabad.*
[2] *Department of CSE, NallaMalla Reddy Engineering College, Hyderabad.*
[3] *Department of ECE, Srinidhi Institute of Science & Technology, Hyderabad*

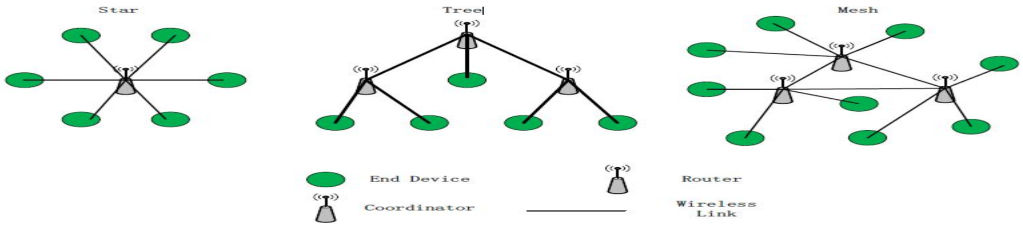


Figure 1 ; Common Topology of WSN in Zigbee.

WSNs can be classified in two categories based on their underlying communication architectures: hierarchical WSN (HWSN) and distributed WSN (DWSN) [8]. In HWSN, all the involved entities can be divided into three tiers. The first tier contains a centralized base station which is relatively powerful and controls the whole network. The second tier consists of several cluster heads. Each cluster head manages a set of sensor nodes and reports to the base station while able to behaving as sensor nodes in the same time. The third tier contains the resource constrained sensor nodes and they are used to collect data and submit the useful information to the cluster head. In DWSN, all the sensor nodes are able to talk to each other directly, so it is more like P2P communication architecture rather than client-server architecture. That means, a sensor node can be sender and responder at the same time, and in order to achieve security requirements, a secure communication channel needs to be established between every pair of sensor nodes. This decentralized property makes key management even harder for DWSN.

1.2 Overview of Internet of Things (IoT) : Unlike the legacy Internet, IoT describes a paradigm where objects are part of the Internet and the whole society is "always connected" [9]. The transition from legacy Internet to IoT starts from Wireless Sensor Networks (WSNs), which allows similar wireless sensors to communicate in order to achieve certain functions [10]. Machine to Machine (M2M) communication [11] extended the model of WSN by introducing embedded intelligence and self-organization based on the wireless networks' capability of delivering broadband data service at a significantly low cost. The networks get more complex both logically and topologically. Even devices which have considerably different capabilities and are located far apart can communicate under the concept of M2M. IoT is a further extension of M2M. It tends to interconnect wider sets of objects as well as achieve universality and interoperability. The devices connected by IoT can be extremely powerful as servers, or be extremely resource constrained RFID tag.

1.3 Security in WSNs and IoT : WSNs and IoT share many similarities. For example; most of the communications are achieved through insecure wireless communication channels, which enable the attackers to capture the transmitted packets very easily.

1.3.1 Security in WSNs : Security is one of the main challenges in WSNs due to the broadcast nature of the communication channel, the limited energy, computational and memory resource they tend to have, and their potential deployment in remote, and physically insecure areas. Certain WSNs become valuable targets to attackers because of their important functions and the sensitive information they are collecting and transmitting. WSNs are vulnerable to many attacks, such as DoS attack, Man-in-the-Middle attack and black hole attack. The security concern is more serious in applications related to military and e-health. In WSNs, the goal of security is to efficiently achieve availability, survivability, scalability, confidentiality and integrity. Authentication, encryption and other techniques and protocols are proposed and widely used in WSN applications in order to cope with various attacks. Using proper encryption techniques to protect the information is fundamental in achieving security in WSNs, and this requires the involved communication entities to have either symmetric keys or asymmetric keys for the encryption/decryption operations. The generating, distributing, exchanging, storage, use and replacement of keys needs to be properly managed using key management schemes. Due to the diversity of WSN technologies/implementations and the plurality of WSN deployment areas abd applications, it is difficult to provide a "one solution fits all" solution in terms of securing transfer of information; the corresponding key management schemes may have to be selected/designed in accordance to the specific nature of deployed sensor nodes, the environment of deployment and the security requirements of the serviced application. Key management schemes can be roughly classified into two groups: asymmetric encryption based and symmetric encryption based. The asymmetric encryption operations are considerably more resource consuming compared to the symmetric ones, and resource constrained sensor nodes cannot perform such complex operations. The symmetric encryption based schemes also have their own limitations, one of the major concerns been that the small size of memory placed on-board of sensor nodes limits the number of symmetric keys a node can store, which further limits the scale of the network. Dustin et al [8] gave a review on the existing key management schemes in WSNs. They classified the key management schemes into three categories: pair-wise key management schemes, random key-chain based key pre-distribution solutions, and network-wise key management schemes.

1.3.2 Security in Internet of Things : Potential application areas of WSNs and IoT, such as military, telehealth and commercial transactions usually involve sensitive, even highly classified information. How to achieve secure communication under the concept of IoT is becoming a primary problem and challenge. This problem is hard to solve because of two reasons. The first is that the vast number of involved objects makes it impossible for every device to become "known" to all the others, thus there is lack of trust among devices and lack of secure channels between them. The second one is that there is significant diversity and dissimilarity between devices in terms of energy availability, computing power and storage space. Thus, highly resource constrained devices cannot run heavy cryptographic algorithms like powerful devices do. To address the first problem, the Host Identity Protocol (HIP) [17] was proposed. It introduced a new name space, which is similar to the name spaces we have in the legacy Internet: Domain Name Service (DNS) and Internet Protocol (IP) addresses. HIP also provides a secured Base Exchange (BEX) mechanism for two devices to agree on a shared key.

1.4 Overview of Secret Sharing : Secret sharing means dividing one secret into pieces and sharing them with a set of shareholders. Each piece of the secret is called a share of the secret. An interesting example of secret sharing application is the storing of the nuclear missile launch code. If the code is possessed by only one individual, the missile can be easily launched by mistake or maliciously. And if this individual is captured or killed, the missile cannot be launched because of the missing of the launch code. So the code should be shared by multiple people, each of them possessing only part of the code. When the code is needed, all these people will gather together and combine all the code parts. This method prevents accidental or malicious launch of the nuclear weapons, however, the problem still remains when someone who possesses partial code is captured or killed. How can we still be able to launch the missile with some parts of the code missing?

The concept of threshold secret sharing solves this problem. It allows the majority of the code owners to recover the rest of the code parts while preventing the minority from launching the missiles accidentally or maliciously. This concept applies the voting mechanism and relies on the judgment of the majorityIn Shamir's scheme, a centralized dealer shares a secret with multiple shareholders; however, any entity, including the shareholders, cannot retrieve the secret until it obtains the help of a certain number of shareholders. Secret sharing schemes with this property are called (***t***,***n***) threshold secret sharing schemes, where ***t*** represents the minimum number of shareholders that are required to retrieve the secret and ***n*** stands for the total number of existing shareholders.

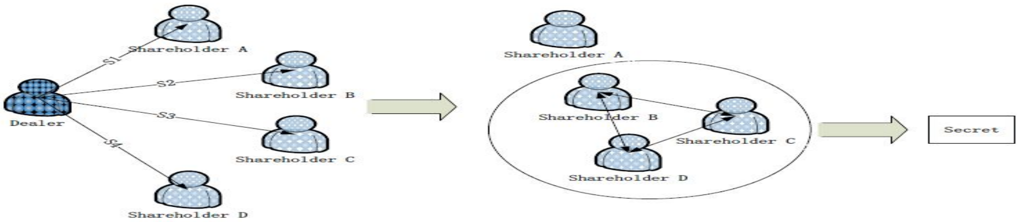


Figure 2 Basic Threshold Secret Sharing

In 1982, Mignotte [20] proposed a new secret sharing scheme that is based on the Chinese Remainder Theorem (CRT). However, with this scheme, the exposure of any individual share can reveal some information of the original secret and narrow down the searching space of the secret, thus it is not theoretically secure. In 1983, Asmuth and Bloom proposed another secret sharing scheme that is also based on CRT but incorporated some improvements. According to, Asmuth-Bloom secret sharing scheme is "asymptotically ideal and perfect zero-knowledge if the parameters of the system satisfy a natural condition", which makes the reveal of secret shares not affect the size of the searching space too much and a minimum searching space is guaranteed. Also indicates that the computational complexity of the secret reconstruction from *t* shares for the Asmuth-Bloom's scheme is O(t)while it is O(t(log^{2}t)) for the Shamir's secret sharing scheme. This makes the Asmuth-Bloom's scheme theoretically more computationally efficient than Shamir's scheme, since the architecture they are using is the same. The improvement comes from the use of different underlying mathematical mechanisms; CRT for Asmuth-Bloom scheme and Lagrange polynomial interpolation for Shamir's scheme.Both, Shamir's scheme and Asmuth-Bloom's scheme can share only one secret by distributing a set of shares once.

1.5 Problem Statement and Motivation : As we can see from the previous sections, many works have been done in the areas of multi-secret sharing, key management and Internet of Things. However, the existing solutions have their own limitations. For the key management schemes in WSNs, most of them may lack flexibility, or cannot provide adequate security, or have some other disadvantages. Nowadays, WSNs have been integrated to the Internet of Things. Also existing and new applications of the WSNs are becoming more complex and complicated. Complex and highly intelligent applications and engineered structures can be manufacturing by combining large number of sensing, processing and actuating nodes of limited capabilities. As for the previously mentioned HIP, which is an identity-based cryptosystem, is implemented using public key encryption technologies. This allows devices to be accessed or authenticated everywhere using their unique identifications. HIP-BEX, provided by HIP, is designed to achieve authenticated key agreement between two HIP peers that have legitimate host identifiers using the Diffie-Hellman key exchange protocol. However, since Host identities in HIP are generated and verified using RSA and HIP-BEX is based on the Diffie-Hellman key exchange protocol [1], their involved heavy cryptographic operations make HIP computationally intensive and energy consuming. Thus HIP is not suitable for use with resource constrained devices. A number of research contributions were made, addressing this problem. HIP Diet key exchange (HIP-DEX) and Lightweight HIP are two examples that proposed to reduce the resource consumption of the key establishment operations. In 2011, Moskowitz et al proposed HIP Diet EXchange (HIP-DEX) to further improve the HIP-BEX by using long term Elliptic Curve Diffie-Hellman (ECDH) public value as HIP hosts identifiers. HIP-DEX requires one pair of DH keys instead of two pairs that are required by HIP-BEX. Lightweight HIP (LHIP), proposed by T. Heer in 2007, only uses hash chains to bind successive messages to provide security of communication. However, LHIP only guarantees the ongoing session is not hijacked, which is minimal security requirement. In 2012, Y. B. Saied and A. Olivereau [12] proposed their schemes to replace the heavy Diffie-Hellman key agreement of HIP-BEX, including this (*t*,*n*) threshold distributed key exchange for HIP, which we refer to as TD-HIP in later discussion. They successfully replaced DH key exchange with lighter public key infrastructure (PKI) and introduced a new collaborative approach to share the heavy workload of the constrained host. In TD-HIP, Shamir's (*t*,*n*) secret sharing scheme [18] is used to allow the newly introduced third entity, proxy, to collaboratively help the resource constrained device during the key agreement process.

II. ARCHITECTURE

The architecture of our proposed key management scheme is hybrid architecture, which means the sensor nodes could communicate using both Peer-to-Peer (P2P) and clustered communication model. And in order to make this hybrid architecture to work efficiently and securely, we introduce dedicated relatively more powerful devices in to the system. Here, we reference these devices as key managers (KMs). However, the key managers can also become the intermediate hops to help a sensor node or another key manager to transmit its encrypted messages to another remote node. And in clustered communication, some key managers might become cluster heads as well. Therefore, after the deployment, all key managers will collaboratively generate an efficient topology for communication and routing purpose, and this topology will keep updating based on the trustworthiness of the key managers and their relative locations.

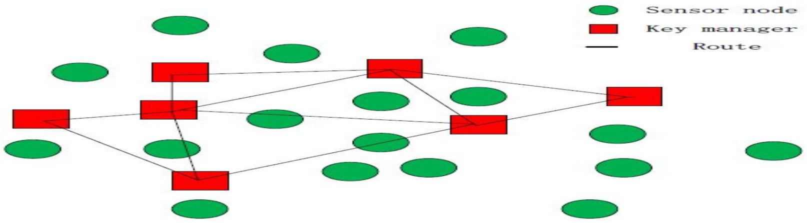


Figure 3: Basic Key Manager and Sensor Node Distribution and Architecture

In Figure 3, key managers and sensors are randomly deployed in the same area which makes them geologically close to each other. The key managers are powerful devices with longer communication range and more resources. Unless it is necessary, we only use key managers to route the packets through the established topology. This will save energy for the resource constrained sensor nodes, and from the system's perspective, using key managers as intermediate hops can save the overall number of hops and the system will be more efficient. After the deployment, every key manager should be able to find more than one key managers within its own communication range, and every sensor should also be able to communicate directly with more than one key managers in case of some manager might be compromised by the attackers or get offline after the deployment.

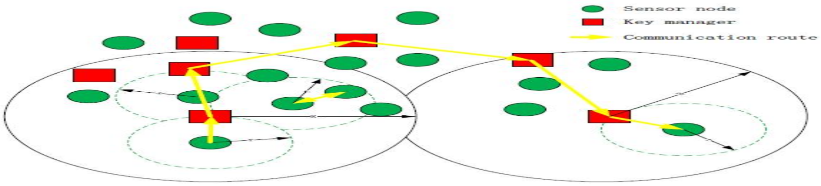


Figure 4: P2P Communication Model

In Figure 4, we use yellow arrows indicate the communication connection between two sensors. There are two types of complete connections: one is the direct connection connecting two geologically close sensor nodes; the other one is multi-hop connection using multiple key managers as intermediate hops.

Figure 5 shows the clustered communication model in our proposed scheme. Sensors could communicate with the key manager which is in their communication range, and the key managers, which have larger communication ranges, could act as routers between two sensors. So, there are two ways of constructing a cluster, the first one is constructing cluster based on the geological location information, the other one is constructing cluster based on the function of the sensor nodes, which means, several remote sensors which have the same function, such as temperature testing, as long as they have the same cluster key, they can be considered as cluster members of the same cluster.

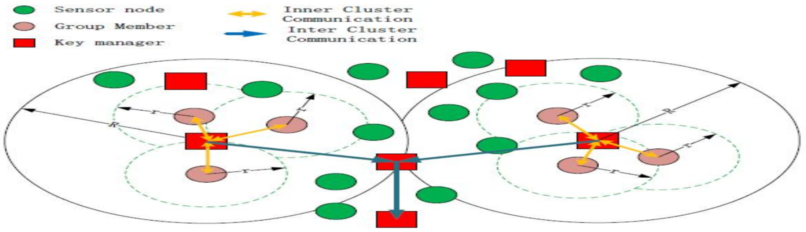


Figure 5: Clustered Communication Model.

In Figure 5, there are two clusters constructed based on the geological location information, each has a key manager as the cluster head, and since the two cluster heads cannot communicate directly because of the communication range, they collect data from their own cluster members and submit to an intermediate key manager, which continuously transmits the collected data to the next hop until the packets arrive the destination.

2.1 Basic Secret Sharing Schemes : We presented two different (***n,t,n***) multi-secret sharing approaches in previous section, one is Lagrange polynomial interpolation based (***n,t,n***) secret sharing scheme and the other one is CRT based (***n,t,n***) secret sharing scheme. They both use proxy to outsource the workload from the highly resource constrained devices. Regarding to our application scenario, the key managers are responsible for the secret shares generation and distribution, the sensors are responsible for the secret reconstruction operations which is energy consuming in both schemes. Since the proxy are the most powerful devices in this application, we propose to use CRT base multi-secret sharing scheme instead of Lagrange polynomial interpolation based one since the theoretical computational complexity for Shamir's scheme is $O(t\log^2 t)$ while it is O(t) for Asmuth-Bloom scheme and the testing results of computational costs of the two schemes also indicate the CRT based scheme is more efficient regarding to the dealer while the two schemes have the similar cost regarding to the reader.

2.2 Notation Used in Key Management Scheme

we list the important notation and the corresponding descriptions in Table 2.

Table 2 Key Management Scheme Related Notation and Corresponding Descriptions

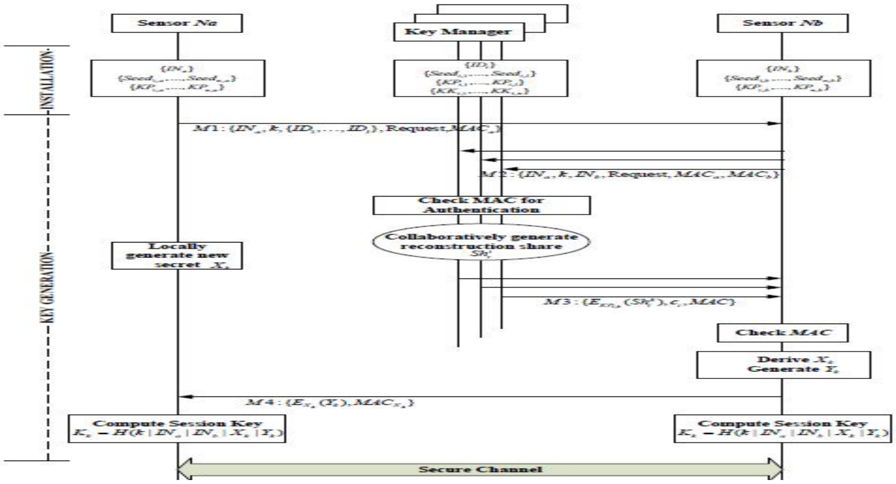
Symbol	**Description**
N_j/ N_p	Sensor nodes with a sequence number of j or q , where , j q ,1,2,...,l, $j \neq q$
KM_i / KM_p	Distributed key manager with a sequence number of , where , and i,p= 1,2,…….,t, t+1……n and i≠p
n	The total number of the key manager
t	Pre-defined threshold
l	The total number of the sensor nodes
ID_i	The initial identification number for i^{th} key manager
IN_j	The identification number for j^{th} node
k	Random number that generated by the sensor and initiates the communication for the current session
S_k	Session key generated in the session tagged with random number k
$KP_{i,j}$	Secret key pre-shared by KM_i and N_j
$KK_{i,p}$	Secret key pre-shared by KM_iandKM_p

$sh^{k}_{i,p}$	Sub-share generated forKM_pbyKM_i, here this is a superscript.
$SH_{i,j}$ / $SH_{p,j}$	Master share, shared between and KM_iandN_j
Sh^{k}_{i}/ Sh^{k}_{p}	Reconstruction shares generated byKM_i / KM_p , here this k is a superscript
IC	Identification for cluster
H(x,y)	One- way function with two inputs
RN^{k}_{i}	Random number generated by the i^th key manager in k^th session, here this k is a superscript.
$Seed_{i,j}$	Secret seed used in hash function shared between KM_j and N_j
m_i	Large prime number which satisfy $m_0 \prod_{i=n-t+2}^{n} m_i < \prod_{i=1}^{t} m_i$, and $0 < m_0 < m_1 < \cdots < m_n$
C_i	$M = \prod_{i=1}^{t} m_i \ , M_i = M \,/\, m_i \ , \ c_i = M_i \times (M_i^{-1} \bmod m_i) \ , \ 1 \le i \le t$.

2.3 Flexible Key Management Scheme : We modified CRT based (***n***,***t***,***n***) multi-secret sharing scheme to allow the decentralized key managers collaboratively generate and assign new session secret keys for the sensors. As long as there are***t*** or more legislate key managers and they are able to communicate with other, the sensors within their communication range can always obtain secret keys for P2P communication or clustered communication.

2.3.1 P2P Communication Mode : As shown in Figure 6, the P2P communication mode has two phases: installation and key generation.

Installation Phase: After the deployment, all the key managers and sensor nodes are randomly distributed in a large area. They will instantly generate the topology for clustering and routing purposes. This topology will be updated periodically.



Sensor Na
Key Manager
Sensor Nb
INSTALLATION
{IN_a}
{Seed_{1,a},...,Seed_{n,a}}
{KP_{1,a},...,KP_{n,a}}
{ID_i}
{Seed_{i,1},...,Seed_{i,j}}
{KP_{i,1},...,KP_{i,j}}
{KK_{i,1},...,KK_{i,n}}
{IN_b}
{Seed_{1,b},...,Seed_{n,b}}
{KP_{1,b},...,KP_{n,b}}
M1 : {IN_a, k, {ID_1,...,ID_t}, Request, MAC_a}
M2 : {IN_a, k, IN_b, Request, MAC_a, MAC_b}
Check MAC for Authentication
Collaboratively generate reconstruction share Sh_i^k
KEY GENERATION
Locally generate new secret X_k
M3 : {E_{KP_{i,b}}(Sh_i^k), c_i, MAC}
Check MAC
Derive X_k
Generate Y_k
M4 : {E_{X_k}(Y_k), MAC_{X_k}}
Compute Session Key
K_k = H(k | IN_a | IN_b | X_k | Y_k)
Compute Session Key
K_k = H(k | IN_a | IN_b | X_k | Y_k)
Secure Channel

Figure 6 : Key Generation

1) Sensor ***S_a***generates a random number for the purpose of distinguish different communication sessions in case of sessions between the same two nodes always get the same session key. Then ***S_a***selects a sub-set of key managers within the network and send the identification numbers to sensor ***S_b***together with the random number and the key setup request through message M1. Sensor ***S_a***also generates HMACs for the message for integrity check and authentication purpose. k_k

Key material is then calculated locally by the following operations: X_k

$RN^{k}_{i}=H(k,seed_{i,a)}$ (1)

$X_k\equiv \sum^{n}_{i\text{-}1}(RN^{k}_{i\,.}\ SH_{i,a})(\text{mos } M)$ (2)

2) Sensor ***S_b***receives the message M1 and if the request is accepted, sensor ***S_b***contacts the key managers appointed by sensor ***S_a***by sending message M2. M2 passes along the content in M1 and adds its own identification number and the HMACs.

3) After receiving message M2, key managers will check the HMACs generated by both sensors to check the integrity of the message, and since the seed used in the HMAC operation is known only by the corresponding sensors and the key managers, only legitimate sensors are able to generate the HMACs correctly.

4) All the key managers collaboratively generate reconstruction shares $Sh^{k}_{i.}$ Using master shares and shared seeds, the key manager can generate sub-shares for on sensor ***S_a***'s behalf:

$sh^{k}_{i,p}\equiv H(k,seed_{i,a})\ .\ SH_{i,a}(\text{mod}m_{p})$ (3)

Upon receiving all the sub-shares from selected key managers, can generate the reconstruction share: KM_i

$sh^{k}_{i,p}\equiv\sum^{n}_{p\text{-}}1sh^{k}_{p,i}(\text{mod}m_{i})$ (4)

KM_ialso computes $c_i$for sensor ***S_b***in order to further reduce the energy consumption.

The reconstruction shares are then encrypted using the unique shared keys shared between key managers and sensor $\boldsymbol{S_b}$, HMACed using the shared seeds, and sent to sensor $\boldsymbol{S_b}$for the key material retrieving.

5) Sensor $\boldsymbol{S_b}$decrypts M3 and uses the reconstruction shares and $c_i$to reconstruct the shared secret, X_k.

$$X_k \equiv \sum_{i=1}^{t} Sh_i^k \cdot c_i (\bmod M) \qquad (5)$$

6) Then sensor $\boldsymbol{S_b}$generates its own key materialY_k, encrypts $Y_k$by X_k.Y_k. is then sent back to sensor $\boldsymbol{S_a}$and both sensors calculate the final session key using the pre-agreed hash function:

$$K_k = H(k \,|\, IN_a \,|\, IN_b \,|\, X_k \,|\, Y_k) \qquad (6)$$

2.3.2 Clustered Communication Mode : Since we are using key managers to collaboratively generate and share secret keys, we could generate one key for just two nodes or a set of nodes. In the first case, two nodes share an exclusive key, noted as $EK^k_{j,q}$, means this key reconstruction is initialized by node $N_j$in the k-th session , with node N_q. In the second case, we say that the key shared by more than two nodes is a group key, denoted as $GK^k_{c,j}$, means this group is initialized by node $N_j$in the k-th session with a cluster identification number c, c= 1,2,3,…, and actually $GK^k_{c,j}$= S_k.Assuming key manager KM_iclaims to be a cluster head and initiates the construction of a new cluster based on the function of the sensor nodes, which means the sensor nodes are not geologically close to each other and KM_i. Assuming the selected cluster members are $N_{m,}$m=1,2,...,l.

There are two methods of constructing a new cluster. Since the key manager KM_ishares a unique secret key with each sensor node, the cluster can be constructed by asking $KM_i$to generate a cluster communication session key and distribute to the selected sensor nodes secretly by encrypting all the messages using the pre-shared keys. Then all the members within the newly created cluster start communicate privately using the distributed secret key.

III. KEY MANAGERS LEAVE OR JOIN THE SYSTEM

Delete Key Manager. : In several occasions, we need to delete a key manager from the system. This key manager might be removed from the targeting location, have used up all the energy, be labeled as untrustworthy based on the feedbacks of the sensors, and so on. To delete a key manager, most of the legitimate key managers need to send an encrypted command to the sensors to inform them the left of a key manager. All the sensors received this command will delete all the pre-shared keys associated with this key manager. This method works because of the previous assumption that most of the key managers are trust worthy.

New Key Managers Join the System. : A new key manager does not share a pre-shared key with all the sensor nodes. If the key manager wants to obtain those pre-shared keys with all the sensor nodes, we propose to use the following procedures. Assuming the newly joined key manager isKM_p, all the other existing key managers are $KM_i$with i=1,2,…,n. KM_ifirst uses public key infrastructure to authenticate the new member and then exploits asymmetric encryption techniques, such as Diffie-Hellman key exchange scheme, to share a symmetric key with KM_p.

Key manager KM_icomputes random numbers as the secret:

$$H(KP_{i,j}\ , Seed_{i,j}\ ,ID_p) = R^p_{i,j} \qquad (7)$$

$ID_p$are known by all the members, but $KP_{i,j}$and $Seed_{i,j}$is only known by $KM_i$and the j-th sensor node, so $R^p_{i,j}$is also known only byKM_i and the j-th sensor node. All KM_ithen encrypt $R^p_{i,j}$and send toKM_p .KM_pcan obtain all the $R^p_{i,j}$with i=1,2,…,n j=1,2,…,l. KM_p will further compute the new pair-wised key using one-way hash function:

$$KM_{p,j} = H(\textstyle\sum^n_{i=1} R^p_{i,j}, ID_p) \, (8)$$

When KM_pneeds to communicate with the j-th sensor node, the sensor will computes $KP_{p,j}$locally using Equation (7) and Equation (8) based on the necessary information.

The first timeKM_p joins the system, it obtains the keys from the other key managers using the method we presented above. All the sensor nodes will be informed about the new key manager, and locally computes the secret key when necessary. During the first communication session between $KM_p$and the sensor node, the sensor node will encrypt a puzzle and send to KM_p, if KM_psuccessfully decrypts the message and provides the correct answer, that means all the other legitimate key managers already authenticated this new key manager and provided $KM_p$the required keys with their trust.

HIP-MEX: New Key Agreement Schemes Based on (*n,t,n*) Multi-Secret Sharing for HIP Based Internet of Things

We first introduce TD-HIP, which is the most efficient existing key agreement scheme for HIP, in detail. Then, we apply the (*n*,*t*,*n*) multi-secret sharing scheme based on Lagrange polynomial interpolation to HIP context. We also propose to modify our CRT-based (***n,t,n***) multi-secret sharing scheme into key agreement scheme in HIP to further reduce the communication and computation workload for the resource constrained devices. The last two schemes we also referred to as Lagrange polynomial interpolation based HIP Multi-secret sharing Key Exchange (HIP-MEX-LPI) and CRT based HIP-MEX-CRT.

3.1 HIP-MEXs : In this section, we present two key exchange schemes for HIP using the same architecture as TD-HIP. They are both based on (*n*,*t*,*n*) multi-secret sharing schemes and according to the different underlying mathematical principle, we refer to them as HIP-MEX-LPI (Lagrange polynomial interpolation based multi-secret sharing key exchange for HIP) and HIP-MEX-CRT (CRT based multi-secret sharing key exchange for HIP).

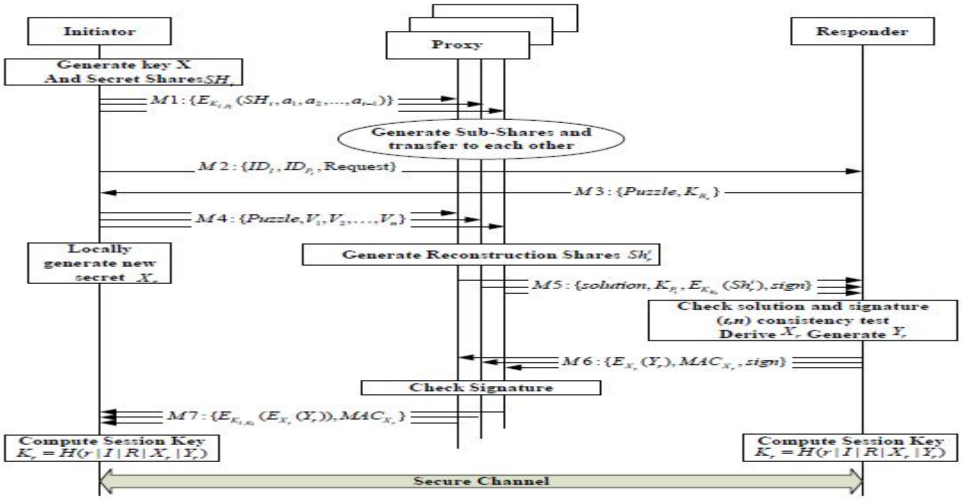

Initiator
Proxy
Responder
Generate key X And Secret Shares SH_i
$M1:\{E_{K_{I,P_i}}(SH_i, a_1, a_2, \ldots, a_{t-1})\}$
Generate Sub-Shares and transfer to each other
$M2:\{ID_I, ID_{P_i}, \text{Request}\}$
$M3:\{Puzzle, K_{R_r}\}$
$M4:\{Puzzle, V_1, V_2, \ldots, V_n\}$
Locally generate new secret X_r
Generate Reconstruction Shares Sh_r^i
$M5:\{solution, K_{P_i}, E_{K_{R_r}}(Sh_r^i), sign\}$
Check solution and signature
(t,n) consistency test
Derive X_r Generate Y_r
$M6:\{E_{X_r}(Y_r), MAC_{X_r}, sign\}$
Check Signature
$M7:\{E_{K_{I,P_i}}(E_{X_r}(Y_r)), MAC_{X_r}\}$
Compute Session Key $K_r = H(r \mid I \mid R \mid X_r \mid Y_r)$
Compute Session Key $K_r = H(r \mid I \mid R \mid X_r \mid Y_r)$
Secure Channel

Figure 7 Lagrange Polynomial Interpolation Based HIP-MEX

HIP-MEX-LPI, as shown in Figure 8 is a direct application of the (*n*,*t*,*n*) multi-secret sharing scheme we reviewed in Chapter 2 proposed by Liu, Yanxiao et al.

This type of HIP-MEX consists of two phases: preparation phase and key generation phase.

Preparation Phase: The preparation phase is active when every (*n-t+1*) secret is generated and exchanged to make sure that the attackers cannot compromise the system.

Message M1: The initiator generates (*t*-1) degree sub-polynomials f_i (x) = SH_i+ a_1x^1 + a_2x^2 + …+ $a_{t-1}x^{t-1}$ mod p for each proxy, and then encrypts these master shares using the pre-installed shared secret key$K_{I,pi}$. Here $SH_i$is the master share for the *i*-th proxy. The cipher texts will be distributed to proxy through message *M1*. Upon receiving the ciphers, the proxy will decrypt them and get their own master share, and using this master share, the proxy $P_i$can generate sub-shares sh^j_i=$f_i(x_j)$ and distribute to proxy $P_j$secretly.

Key Generation Phase:

Message M2: The initiator sends request to the responder together with its own id and the most trusted proxy's ID.

Message M3: The responder will verify the identity of the initiator, and if the responder accepts the request, it will send a response message containing its public key $K_{RT}$and a puzzle back to the initiator. K_{Rr}

Message M4: The initiator receives message M3 and then broadcast the generated vectors{ V_1,V_2,…V_n} and the received puzzle to all its proxy.

Message M5: The proxies compute the solution for the responder and they then can use the received vectors and previously obtained sub-shares to locally generate the reconstruction shares:

Sh^i_r=(V_1·sh^1_i + V_2·sh^2_i+ ···+V_n·sh^n_i)mod p

The reconstruction shares will be secretly sent to the responder through message M5.

Message M6: The responder will verify the solution and the signed signature first, and then check the (*t,n*) consistency of the received reconstruction shares. If everything checks out, the responder will be able to derive the secret using $X_r = \sum_{j=1}^{t} Sh_r^j \left(\prod_{r=1, r\neq j}^{t} \frac{-x_r}{x_j - x_r} \right) (\bmod p)$ Then the responder generates key material $Y_r$and encrypts$Y_r$usingX_r.

Message M7: The proxy will verify the authentication and integrity of the received information $E_{Xr}(Y_r)$, and then encrypt the received cipher using the secret key shared with the initiator. The initiator can decrypt all the information delivered through message M7, which means the initiator and the responder exchanged their key material and will be able to generate a session key for further communication.

3.2 HIP-MEX-CRT : Figure 8 shows the proposed HIP-MEX-CRT in detail. As we can see, the proposed scheme has a similar architecture to HIP-MEX-LPI. However, by using CRT-based multi-secret sharing, HIP-MEX-CRT can finish key exchange with minimum interaction between the initiator and the other entities; most of the heavy workload and expensive cryptographic operations are outsourced to the relatively powerful devices.

Table 3 lists all the useful notation and their corresponding descriptions used to describe HIP-MEX-CRT

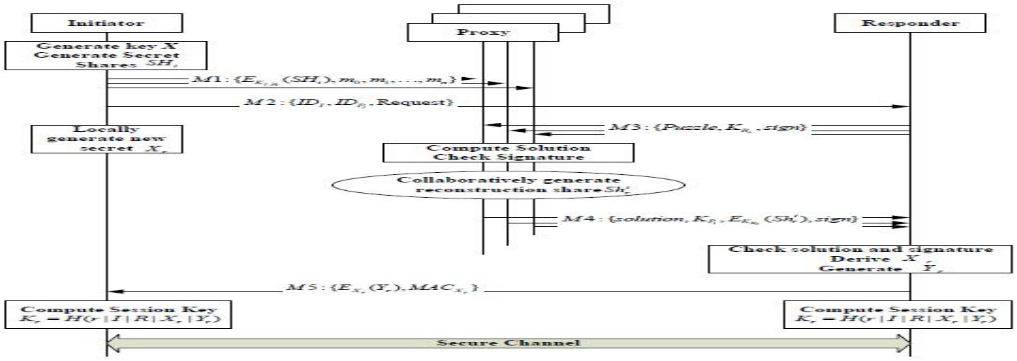

Initiator
Proxy
Responder
Generate key X
Generate Secret Shares SH_i
$M1: \{E_{K_{I,P_i}}(SH_i), m_0, m_1, \ldots, m_n\}$
$M2: \{ID_I, ID_{P_i}, \text{Request}\}$
$M3: \{Puzzle, K_{R_r}, sign\}$
Locally generate new secret X_r
Compute Solution
Check Signature
Collaboratively generate reconstruction share Sh_r^i
$M4: \{solution, K_{P_i}, E_{K_{R_r}}(Sh_r^i), sign\}$
Check solution and signature
Derive X_r
Generate Y_r
$M5: \{E_{X_r}(Y_r), MAC_{X_r}\}$
Compute Session Key
$K_r = H(r \mid I \mid R \mid X_r \mid Y_r)$
Compute Session Key
$K_r = H(r \mid I \mid R \mid X_r \mid Y_r)$
Secure Channel

Figure 8 Proposed HIP Multi-Secret Sharing Based Key Exchange (HIP-MEX) Scheme

HIP-MEX-CRT is proposed to further reduce the workload on extremely resource constrained devices. Instead of distributing secret shares or vectors every time, HIP-MEX-CRT proposes to distribute key materials once for all the key inquiries in a long period. Within the same period, the session key will be derived from the same key materials, and each session key will be unique and independent from another. Every session key will have its own expiration time, which will further enhance the security of HIP-MEX-CRT. HIP-MEX-CRT can be divided into several phases, and we will give a detailed description of each phase in the following discussion.

Installation Phase : There are two ways to assign proxy to the initiator. The first one is to privately assign a secret symmetric key to each pair of the devices before the deployment of the system, so initiator can “recognize” its proxy through the verification of the knowledge of the pair-wise secret key. The second one is more flexible and costly. A trusted third party will assign a HIP host identity to all the devices. After the deployment, the initiator can use the HIP protocol to authenticate each other and then use the Diffie-Hellman key exchange scheme to share a secret key. The Diffie-Hellman key exchange is only performed a limited number of times. Here, in order to make the system simple, we propose to use the first way for an initiator to select and communicate with the proxy. The initiator randomly selects a large positive integer ***X*** as the master key. In the meantime, it secretly selects random large prime numbers .$m_0, m_1, m_2, \ldots, m_n$.

Then, the initiator generates the shares of the master key ***X***; those shares will be used as key materials later.

$$SH_i \equiv X\square(\mathrm{mod} m_i) \qquad (9)$$

Master shares (m_i, SH_i) with i=1,2,…,n will be encrypted using symmetric encryption techniques. Here we propose to use AES-256 to provide sufficient secrecy and efficiency.

Then the initiator constructs the ***M1*** message.

Initialization Phase : When the whole system starts to operate, the initiator initiates a new communication session with responder $\boldsymbol{R_r}$by sending the inquiry message ***M2***. Message ***M2*** contains the HIP host identities of the initiator and all the proxy selected. And it also contains the inquiry command to initiate a new communication session as well as the underlying communication protocol and security techniques. Upon receiving ***M2***, the responder needs to decide whether it wants to communicate with the initiator or not. If it does not, the responder will just stop responding and the initiator will wait until the request times out. If it does, the responder will generate message ***M3*** for each proxy selected and informed by the initiator. The *Puzzle* is generated distinctly for every proxy, and the signature *sign* is generated by the responder’s private key which is paired with the public key K_{Rr}. So the proxy will be able to authenticate the responder on behalf of the initiator.

Key Material Generation Phase : The key is generated in parallel by both the initiator and the responder using different methods. The initiator avoids most communicational and cryptographic operations; it computes the partial secret key locally based on the key shares distributed during installation.

$$X\square_r \equiv (\textstyle\sum^n{}_{i=1} SHi \cdot h_{i,r})(\mathrm{mod} M) \qquad (10)$$

$$X_r \equiv X\square_r\ (\mathrm{mod} m_0) \qquad (11)$$

$X_r$is the partial secret and will be kept, so in this phase, the initiator will not generate message.

The proxies only have part of the key generation materials, and they will help the responder to finally derive the secret. The proxy $\boldsymbol{P_i}$will first check the signature created by the responder to authenticate the received message, and then compute the solution for the received *Puzzle*;

$Sh^j_{i,r} \equiv SH_i \cdot h_{i,r}(\mathrm{mod} m_j)$ (12)

$Sh^j_{i,r}$are the sub-shares generated by $P_i$in the *r*-th session and will be sent to P_j. Messages sent between proxy are not shown in Figure 40 because of the limitation of the space of the figure. However, they are necessary for completing the whole process.After the proxy have received all the sub-shares, they will then add all their received sub-shares together to generate the reconstruction share, and for proxy P_i, the reconstruction share would be as follows:

$Sh^j_{i,r} \equiv (\sum^n_{j=1} sh^i_{j,r})(\mathrm{mod} m_i)$ (13)

Each proxy will construct their own message ***M4***, and deliver it to the responder with their own signature.

IV. KEY RECONSTRUCTION PHASE

The responder performs the key reconstruction after it received all the ***M4*** messages. It will decrypt all the messages first, and then check the solutions and the signatures. If all the solutions and signatures match, the responder will then select ***t*** reconstruction shares randomly and derives the partial key X_r.

Assuming the selected shares are $\{sh^1_r, sh^2_r, \ldots, sh^t_r\}$, let $M_r = \sum^t_{i=1} m_i, M_{ir} = M_r/m_i$,

$y_{i,r} = M^{-1}_{i,r}(\mathrm{mod}\ m_i)$. Then, based on CRT, can compute the unique solution:

$X□_r \equiv (\sum^t_{i=1} sh^i_r \cdot M_{i,r} \cdot y_{i,r})(\mathrm{mod}\ M_r)$ (14)

$X_r \equiv X□_r\ (\mathrm{mod} m_0)$ (15)

Then the responder will randomly select a new large positive integer $\boldsymbol{Y_r}$, and use this new integer as the other partial key. Then the responder will generate message ***M5*** and send it to the initiator directly.

Since the legitimate proxy have already authenticated the responder on behalf of the initiator, if the responder possesses the knowledge of $\boldsymbol{X_r}$, the initiator will consider the responder legitimate.

$K_r = H(r \mid I \mid R \mid Xr \mid Yr)$ (16)

So, both entities are able to encrypt all their further communications using symmetric encryption techniques and the session key Kr. Here, we also propose to use AES-256.

4.1 Comparison and Performance Analysis : In this section, we will compare the three schemes in terms of the computation and communication energy costs of the initiator under the following assumptions:

- Communication protocol is IEEE 802.15.4 (Zigbee) with four operating modes: transmit, receive, listen and sleep. The power consumption for different modes is also different.
- The initiator will switch to deep sleep mode after finishing the transmission/reception
- We use TelosB sensor which has 16-bit MSP430 microcontroller running under 4MHz. The claimed data rate during transmission is 250kbps.
- All three presented schemes use the same underlying cryptographic techniques to generate random numbers and encrypt/decrypt messages.

The power consumption of TelosB at 4MHz with a transmission power of -5DBM is shown in Table 4.

Table 4 Power Consumption of TelosB 5dBM 4MHz with A Transmission power of

Power Mode	
Transmit	54 mW
Receive	61 mW
Listen	60 mW
Sleep	35 W
Compute	4.8 mW

V. CONCLUSION AND FUTURE WORK

We introduced the basics of WSNs and IoT, the security and privacy preserving issues in these two areas, as well as an overview of secret sharing schemes used to share and distribute secrets securely. The computational costs of our scheme have been carefully simulated and tested using C and OpenSSL. The results show that under expected system settings, the proposed scheme successfully offloaded more workloads comparing to the previous works. However, in order to adapt this scheme to various applications, further security enhancements and performances improvements should be done as well. We presented our flexible WSN key management scheme based on (*n,t,n*) multi-secret sharing scheme. All the sensor nodes within this system only need to remember information associated with several key managers, and the key managers will provide the sensors communication session keys after deployment according to the requirements of applications. This scheme is designed for heterogeneous WSNs with two types of communication architecture: distributed and hieratical. While more powerful and replaceable key managers are responsible for the most communication and computation costs, resource constrained sensors offload most of the workload and can work more efficiently. As Internet of Things becomes more and more popular and available around world, security and energy efficiency are two important issues. We present two new key agreement schemes for HIP based IoT using a distributed approach. Those two schemes are built on LPI-based (*n*,*t*,*n*) multi-secret sharing and CRT-based (*n*,*t*,*n*) multi-secret sharing, respectively. While the LPI-based key agreement scheme failed to improve the energy efficiency, the CRT-based key agreement scheme successfully reduced the energy cost for both computation and communication operations. In future work, we will further strengthen the security of the proposed HIP-MEX-CRT.

REFERENCE

[1]W. Diffie and M. Hellman, "New directions in cryptography," *Information Theory, IEEE Transactions,* pp. 644-654, 22 6 1976.

[2] J. Yick, B. Mukherjee and D. Ghosal, "Wireless Sensor Network Survey," *Computer networks,* vol. 52, no. 12, pp. 2292-2330, 2008.

[3] K. Maraiya, K. Kant and N. Gupta, "Wireless Sensor Network: A Review on Data Aggregation," *International Journal of Scientific & Engineering Research,* vol. 2, no. 4, pp. 1-6, 2011

[4] I. Howitt and J. Gutierrez, "IEEE 802.15. 4 low rate-wireless personal area network coexistence issues," *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE,* vol.3, pp. 1481-1486, 16-20 March 2003.

[5] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta and Y. F. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards," *Computer communications,* vol. 30, no. 7, pp. 1655-1695, 2007.

[6] ANT Wireless, Dynastream Innovations Inc, [Online]. Available: http://www.thisisant.com.

[7] S. Adibi, "Link Technologies and BlackBerry Mobile Health (mHealth) Solutions: A Review," *Information Technology in Biomedicine, IEEE Transactions on,* pp. 586-597, 2012.

[8] M. Dustin, J. Shankarappa, M. Petrowski, H. Weerasinghe and H. Fu, "Analysis of key management in wireless sensor networks," *Electro/Information Technology, 2007 IEEE International Conference on,* pp. 263-271, 2007.

[9] L. Coetzee and J. Eksteen, "The Internet of Things-promise for the future? An introduction," *IST-Africa Conference Proceedings, 2011,* pp. 1-9, 2011.

[10] V. Potdar, A. Sharif and E. Chang, "Wireless sensor networks: A survey," *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on.,* pp. 636-641, 2009.

[11] G. Wu, S. Talwar, K. Johnsson, N. Himayat and K. D. Johnson, "M2M: From mobile to embedded internet," *Communications Magazine,IEEE,* vol. 49, no. 4, pp. 36-43, 2011.

[12] Y. Ben Saied and A. Olivereau, "HIP Tiny Exchange (TEX): A distributed key exchange scheme for HIP-based Internet of Things," *Communications and Networking (ComNet), 2012 Third International Conference on,* pp. 1-8, 2012.

[13] L. Atzori, A. Iera and G. Morabito, "The Internet of Things: A survey," *Computer networks,* vol. 54, no. 15, pp. 2787-2805, 2010.

[14] F. Villanueva, D. Villa, F. Moya, M. Santofimia and J. Lopez, "Internet of Things Architecture for an RFID-Based Product Tracking Business Model," *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on,* pp. 811-816, 4-6 July 2012.

[15] D. Manivannan and P. Neelamegam, "WSN: key issues in key management schemes—a review," *Research Journal of Applied Science, Engineering and Technology,* vol. 4, no. 18, pp. 3188-3200, 2012.

[16] N. Liu, J. Chen, L. Zhu and J. Zhang, "A key management scheme for secure communications of advanced metering infrastructure in smart grid," *Industrial Electronics, IEEE Transactions on,* vol. 60, no. 10, pp. 4746-4756, 2013.