Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 8 (2013) pp. 033-057

Analysis and Formal Security Verification of Access Control Schemes in Wireless Sensor Networks: A Critical Survey

Santanu Chatterjee[1], Ashok Kumar Das[2] and Jamuna Kanta Sing[3]

[1]Research Center Imarat
Defence Research and Development Organization, Hyderabad 500 069, India
santanu.chatterjee@rcilab.in

[2]Center for Security, Theory and Algorithmic Research
International Institute of Information Technology, Hyderabad 500 032, India
iitkgp.akdas@gmail.com, ashok.das@iiit.ac.in

[3]Department of Computer Science and Engineering
Jadavpur University, Kolkata 700 032, India
jksing@ieee.org

Abstract: **In an access control scheme, a deployed sensor node proves its identity to its neighbor nodes through authentication and also proves that it has the proper right to access the sensor network. After successful authentication, the shared secret keys should be established between a deployed sensor node and its neighbor nodes to protect communications. In a wireless sensor network, we often require deployment of new nodes to extend the lifetime of the network because sensor network may be lost due to power exhaustion problem or malicious nodes. In order to protect malicious nodes from joining the sensor network, access control mechanism becomes a major challenge in the design of sensor network protocols due to resource limitations of sensor nodes. Until now, there have been ample of access control schemes published in the literature, and each published scheme has its own merits and demerits. In this paper, we have identified all the functionality features and security requirements which must be satisfied for an ideal access control scheme. We have presented and discussed the recently proposed access control schemes available so far in the literature and their cryptanalysis. We have critically analyzed the storage, communication, computational overheads requirement, functionality and security analysis of the existing schemes. Further, we have performed formal security analysis of existing schemes using the widely-accepted AVISPA (Automated Validation of Internet Security Protocols and Applications) tool. All the schemes are vulnerable to different attacks except the Zhou et al.'s scheme and the Chatterjee et al.'s scheme. However, the Zhou et al.'s scheme requires high storage, communication and computational costs. Hence, we feel that there is a strong need to design an ideal efficient access control scheme in future, which should meet all the security requirements and achieve all the functionality features.**

Keywords: Wireless sensor networks, Access control, Key establishment, Authentication, Security.

I. Introduction

In a wireless sensor network (WSN), a large number of small computing nodes, called sensors or motes, are scattered in an area (called the deployment field or target field) for the purpose of sensing important information and transmitting those sensing information to the nearby *base stations* for further processing. Sensor nodes are generally deployed densely in a close proximity to the phenomenon to be monitored. A sensor node is a node in a WSN that is capable of performing some processing, gathering sensory information and communicating with other connected sensor nodes in that network. Sensor nodes communicate by short range radio communications. The base station is a computationally well-equipped node in the network, whereas the sensor nodes are resource-starved. The sensor nodes are usually scattered in a *sensor field* (i.e., deployment area or target field) and each of the scattered nodes has also the capabilities to collect data and route data back to the base station via a multihop infrastructure-less communication through other sensor nodes.

Topology of WSN is dynamic in nature because radio range and network connectivity changes with time. Moreover, sensor nodes may expire due to battery-energy consumption and also new sensor nodes may be needed to deploy to the network in order to replace battery-exhausted nodes and malicious nodes. Further, WSNs are more resource-constrained, denser and may suffer (or take advantage) from redundant information. There are two types of WSN architectures available for wireless sensor networks: first one is the hierarchical architecture and the other is the distributed (homogeneous) architecture.

In a *hierarchical wireless sensor network (HWSN)* shown in Figure 1, there is a hierarchy among the nodes based on their

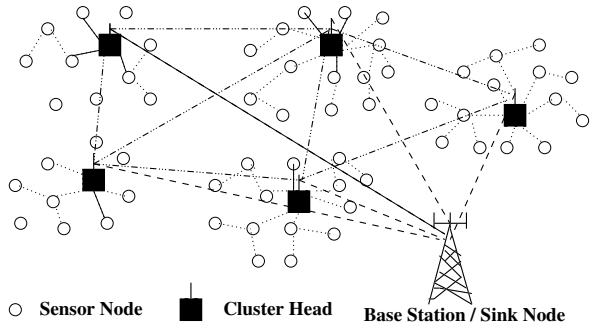

Figure. 1: A hierarchical wireless sensor network (HWSN) architecture.

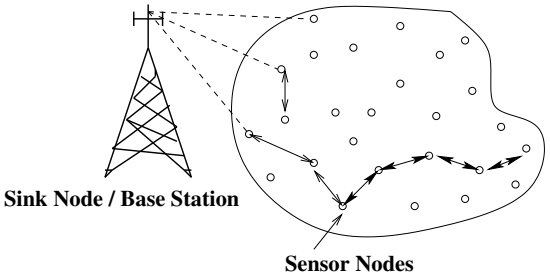

Figure. 2: A distributed wireless sensor network (DWSN) architecture.

capabilities: base stations, cluster heads and sensor nodes. *Sensor nodes* are inexpensive, have limited capability and are generic wireless devices. Each sensor has limited battery power, memory size and data processing capability and short radio transmission range. Sensor nodes in a cluster (group) communicate among each other in that cluster and finally communicate with the cluster head. *Cluster heads* are more resource-rich than sensors. They may be equipped with high power batteries, larger memory storage, powerful antenna and data processing capabilities, and they can execute relatively complicated numerical operations than sensors and have much larger radio transmission range. Cluster heads can communicate with each other directly and relay data between its cluster members and the base station. Finally, a *base station* (also called the *sink node*) is typically a gateway to another network, which is treated as a powerful data processing/storage center, or an access point for human interface. The base station then collects sensor readings, performs costly operations on behalf of sensor nodes and manages the network. In most applications, the base station is assumed to be trusted. As a result, the base station can also be used as key distribution center. Sensor nodes are deployed around one or more hop neighborhood of the base station. Since the base station can reach all the sensor nodes in a network, depending on the applications, the base station can be located either in the center or at a corner of the network. The data flow in such networks are of three types: pairwise (unicast) among sensor nodes, group-wise (multicast) within a cluster of sensor nodes, and network-wise (broadcast) from base station to sensor nodes.

On the other hand, in a *distributed wireless sensor network (DWSN)* shown in Figure 2, there is no fixed infrastructure and network topology is not known prior to deployment of the sensor nodes in the target field. The sensor nodes are usually deployed all over the target area randomly and after deployment sensor nodes form an infrastructure-less multihop wireless communication between them and data is routed back to the base station. The data flow in DWSN is similar to data flow in HWSN with a difference that network-wise (broadcast) flow takes place for every sensor node in the network. More details on survey on sensor networks can be found in [4].

There are several applications of sensor networks. In many applications, such as target tracking, battlefield surveillance and intruder detection, WSNs often operate in hostile and unattended environments. Therefore, there is a strong need for protecting the sensing data and sensing readings. In wireless environments, an adversary not only can eavesdrop the radio traffic, but also has the ability to intercept or interrupt the exchanged messages. Thus, many protocols and algorithms do not work in hostile environments without adequate security measures. Hence, security becomes one of the major concerns when there are potential attacks against sensor networks.

Security requirements in WSNs are very similar to those of ad-hoc networks. WSNs have the following general security requirements [12]:

- *Authentication:* Authenticating other sensor nodes, cluster heads, and base stations before granting a limited resource, or revealing information
- *Integrity:* Ensuring that the message or the entity under consideration must not be altered.
- *Confidentiality:* Providing privacy of the wireless communication channels in order to prevent false reports injection.
- *Availability:* Ensures that the desired network services are available even in the presence of denial of service attacks.
- *Non-repudiation:* Preventing malicious nodes to hide their activities.
- *Authorization:* Ensures that only the sensor nodes those who are authorized can be involved in providing information to network services.
- *Freshness:* Ensures that the data is recent and no adversary can replay old messages.

As suggested in [64], apart from these security requirements, the forward and backward secrecy can be considered when new sensors be deployed in the network and old sensors may fail due to energy problems.

- *Forward secrecy:* When a sensor node leaves the network, it must not read any future messages after its departure.
- *Backward secrecy:* When a new deployed node joins in the network, it must not read any previously transmitted message.

To provide the above security requirements, the key predistribution method has been popularly used in the literature.

In this method, the practical approach is to preload a set of keying information before the deployment of sensor nodes in the target field. After deployment, they discover their neighbor nodes and then establish the secret keys between them using the preloaded keying information. The simplest solution is deterministic approach which uses a single mission master key for the entire network. In this case, in the key pre-distribution phase each node is given the same mission key before deployment in the network. After deployment, in the key establishment phase any two neighbor nodes can communicate securely with each other using that key. However, the main drawback of this simple approach is that the compromise of even a single node in a network would reveal the secret key and thus allow decryption of all network traffic. Another solution of this approach is to use a single shared network-wide key to establish session keys between any two neighbor nodes during the key establishment phase, and then erase the network-wide key. However, the main difficulty of such a variant of the key establishment procedure is that it does not allow deployment of new nodes after the initial deployment in the network.

Other way to provide secure communication with the help of random key pre-distribution approach. Eschenauer and Gligor in 2002 first proposed a random key pre-distribution scheme [26], which consists of the following three phases. In the *key pre-distribution phase*, the (key) setup server (usually the base station) selects a large key pool consisting of randomly generated symmetric keys. Each key is assigned a unique identifier in the key pool. The setup server then chooses a random subset of smaller size from the pool, called the key ring and loads this key ring into its memory before its deployment. In *direct key establishment phase* (also called the shared key discovery phase), each sensor node locates all its physical neighbors within its communication range. In order to establish a secret pairwise key between two neighbor nodes, they exchange the key ids from their key rings. If there is a common key id between their key rings, the corresponding key is taken as the secret key between them. Later, they use this established key for their future secure communication. Nodes which discover that they have a shared secret key in their key rings then verify that their neighbor actually holds the key through a challenge-response protocol. The *path key establishment phase* is an optional phase, and if executed, adds to the network connectivity of the network. Suppose two neighbor nodes u and v fail to establish a secret key between them during the direct key establishment phase, but there exists a secure path between them. Once such a secure path is discovered, u generates a new random key k and securely transmits it along this path to the desired destination node v. In this way, u and v can communicate secretly and directly using k. However, the main problem is that the communication overhead increases significantly with the number h of hops. For this reason, in practice, h is restricted to a small value. When the key pool size is chosen smaller, this scheme provides high network connectivity, that is, any two neighbor nodes can establish a secret key using their key rings with high probability. On the other hand, if some nodes are compromised by an attacker, the probability of compromising a secure link between any two neighbor non-compromised nodes is also high since the key pool size is smaller, and as a result the resilience against node capture becomes poor. Some improved alternatives to the path key establishment have been proposed in the literature [18], [11]. They provide better trade-offs between overheads, network connectivity and resilience against node capture as compared to those for the path key establishment. After that several improvements on the basic random key distribution scheme have been proposed in the literature, some of them are [8], [45], [24], [16].

Several symmetric key pre-distribution and authentication protocols have been proposed to protect sensor networks [26], [8], [45], [24], [13], [25], [14] (see surveys [12], [64], [69] for details). These protocols can establish symmetric pairwise secret keys between neighbor nodes in the sensor network with simple computations and they can reduce the risk of entire sensor network. However, most of the protocols can not be easily implemented as a dynamic access control because the existing old keys as well as broadcasting messages of existing nodes may be updated once new nodes are deployed in the network.

Access control in sensor networks is a mechanism which allows new nodes to join the sensor network dynamically, and key establishment is also included in their access control schemes to help the new nodes to establish shared keys with neighbor nodes so that they can communicate securely in future using the established keys.

A. Our Contributions

Research in access control for sensor networks has received a little attention. Few works [72], [35], [32], [38], [33], [9] are available in the literature to address the access control problem. Though there is another survey [28] on access control and user authentication issues in WSNs, we expect that this paper will provide a deep understanding of access control mechanisms in WSNs. In this survey, our contributions are outlined below:

- We have defined the threat model under which the access control schemes are analyzed for security requirements.
- We have identified the functionality and security requirements under which existing access control schemes are evaluated.
- We have provided a taxonomy of different security protocols in WSNs available in the literature.
- We have described the existing access control schemes and their security drawbacks.
- We have thoroughly analyzed communication cost, computational cost and storage requirement for all existing schemes through quantitative analysis and formulated result.
- We have analyzed the existing access control schemes for formal security verification using the widely-accepted AVISPA model checkers in order to verify whether they are safe against replay and man-in-the-middle attacks.

- Finally, we have compared overall performances of existing schemes and then we have identified that there is a strong need to design an ideal access control scheme in future, which should meet all the security requirements and achieve all the functionality features.

B. Organization of the Paper

The rest of this paper is organized as follows. In Section II, we briefly discuss the mathematical background needed to review the existing access control protocols in sensor networks. In Section III, we discuss the access control problem in wireless sensor networks and then provide the security requirements as well as functional requirements needed for designing an ideal access control scheme. In Section IV, we provide the threat model used in evaluating the security aspects of access control schemes. Taxonomy of different security schemes is given in Section V. In Section VI. we have reviewed in detail the different existing access control schemes and their security drawbacks. In Section VII, we have analyzed different access control schemes for their formal security verification using the widely-accepted AVISPA tool. In Section VIII, we have thoroughly analyzed the existing access control schemes with respect to the security requirements as well as functional requirements needed for designing an ideal access control scheme. Finally, we have concluded the paper in Section IX.

II. Mathematical Preliminaries

A. Elliptic Curve over Finite Field

Let a and $b \in Z_p$, where $Z_p = \{0, 1, \ldots, p-1\}$ and $p > 3$ be a prime, such that $4a^3 + 27b^2 \neq 0 \,(\mathrm{mod}\, p)$. A non-singular elliptic curve $y^2 = x^3 + ax + b$ over the finite field $GF(p)$ is the set $E_p(a, b)$ of solutions $(x, y) \in Z_p \times Z_p$ to the congruence

$$y^2 = x^3 + ax + b \,(\mathrm{mod}\, p),$$

where a and $b \in Z_p$ are constants such that $4a^3 + 27b^2 \neq 0 \,(\mathrm{mod}\, p)$, together with a special point $\mathcal{O}$ called the point at infinity or zero point.

The condition $4a^3 + 27b^2 \neq 0 \,(\mathrm{mod}\, p)$ is the necessary and sufficient to ensure that the equation $x^3 + ax + b = 0$ has a non-singular solution [50]. Otherwise, if $4a^3 + 27b^2 = 0 \,(\mathrm{mod}\, p)$, then the corresponding elliptic curve is called a singular elliptic curve. Let $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be two points in $E_p(a, b)$. Then $P + Q = \mathcal{O}$ implies that $x_Q = x_P$ and $y_Q = -y_P$. We have $P + \mathcal{O} = \mathcal{O} + P = P$, for all $P \in E_p(a, b)$. More precisely, a well-known theorem due to Hasse asserts that the number of points on $E_p(a, b)$, which is denoted by $\#E$, satisfies the following inequality [58]:

$$p + 1 - 2\sqrt{p} \leq \#E \leq p + 1 + 2\sqrt{p}.$$

In other words, an elliptic curve $E_p(a, b)$ over Z_p has roughly p points on it. In addition, $E_p(a, b)$ forms an abelian group or commutative group under addition modulo p operation.

1) Point Addition on Elliptic Curve over Finite Field

Let G be the base point on $E_p(a, b)$ whose order be n, that is, $nG = G + G + \ldots + G(n\,times) = \mathcal{O}$. If $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be two points on elliptic curve $y^2 = x^3 + ax + b \,(\mathrm{mod}\, p)$, with $P \neq -Q$, then $R = (x_R, y_R) = P + Q$ is computed as follows ([58], [40]):

$$x_R = (\lambda^2 - x_P - x_Q)(\mathrm{mod}\, p),$$
$$y_R = (\lambda(x_P - x_R) - y_P)(\mathrm{mod}\, p),$$
$$\text{where } \lambda = \begin{cases} \frac{y_Q - y_P}{x_Q - x_P} \,(\mathrm{mod}\, p), \text{if } P \neq Q \\ \frac{3{x_P}^2 + a}{2y_P} \,(\mathrm{mod}\, p), \text{if } P = Q. \end{cases}$$

2) Scalar Multiplication on Elliptic Curve over Finite Field

In elliptic curve cryptography, multiplication is defined as repeated additions. For example, if $P \in E_p(a, b)$, then $5P$ is computed as $5P = P + P + P + P + P \,(\mathrm{mod}\, p)$.

B. RSA vs. ECC

Watro et al. in [65] proposed a user authentication protocol, called the TinyPK, which uses RSA [54] to authenticate external users and Diffie-Hellman [21] over DLP (discrete logarithm problem) to establish shared keys between external users and sensor nodes in the network. TinyPK uses a public exponent $e = 3$ for computational simplicity. A 1024-bit modular exponentiation with $e = 3$ on MICA1 motes [1] requires 14.5 s. The evaluation of a 1024-bit modular exponentiation for the DLP of the form 2^x, where x is at least 160 bits, requires more than 50 s [65], [48] on both MICA1 and MICA2 motes [1]. Gura et al. in [29] implemented the assembly language for ECC (elliptic curve cryptography) and RSA on the Atmel ATmega 128 processor [1] and they showed in their implementation that a 160-bit point multiplication of ECC requires 0.81 s, whereas 1024-bit RSA public key operation and private key operation require 0.43 s and 10.99 s, respectively.

Compared to RSA, ECC can achieve the same level of security with smaller key size. For example, 160-bit ECC provides comparable security to 1024-bit RSA and 224-bit ECC provides comparable security of 2048-bit RSA [61]. It was pointed out in [7] that in wireless sensor networks, the transmission energy consumption rate approximately over three orders of magnitude greater than the energy consumption rates for computing. However, currently there exist few transceivers with lower communication for transmission and receiver energy consumption. An example of such transceiver is CC2420 [2]. The packet size and the number of packets in transmission will play a crucial role in the performance while designing an access control protocol in sensor networks.

C. Elliptic Curve Discrete Logarithm Problem

Let $E_p(a, b)$ be an elliptic curve modulo a prime p. Given two points $P \in E_p(a, b)$ and $Q = kP \in E_p(a, b)$, for some positive integer k, where $Q = kP$ represent the point P on elliptic curve $E_p(a, b)$ be added to itself k times. Then the elliptic curve discrete logarithm problem (ECDLP) is to determine k given P and Q. It is computationally easy to calculate Q given k and P, but it is computationally infeasible to determine k given Q and P, when the prime p is large.

III. Access Control in WSNs

In order to extend the lifetime of the existing sensor network, deployment of new nodes is necessary. In critical applications including battlefield scenarios, adversaries may directly deploy malicious nodes or manipulate existing nodes to introduce malicious new nodes in the network. To prevent malicious nodes from joining the sensor network, access control is very essential in the design of sensor network protocols. Deployed new nodes, however, may not be always legitimate ones. It may be possible that a malicious node can be directly deployed by an adversary. Those malicious nodes may be indistinguishable from legitimate new nodes, and thus they may be accepted by other legitimate sensor nodes as legitimate ones. As pointed out in [72], in order to prevent malicious nodes from joining sensor networks, access control must be enforced to control sensor node deployment. An access control needs to accomplish the following two tasks:

- *Node authentication:* Through authentication a deployed node needs to prove its identity to its neighbor nodes and also to prove that it has the right to access the sensor network.
- *Key establishment:* Shared keys must be established between a deployed node and its neighbor legitimate nodes to protect communications, after successful authentication between them.

In most applications, WSNs often operate in unattended hostile environments. As a result, there are several potential threats by an attacker or adversary. The following are the potential threats:

- *Sybil attack:* In the sybil attack [49], [23], a malicious node illegitimately takes on multiple identities. Thus, the impersonated identities may belong to existing nodes or non-existing nodes. These malicious nodes may be deployed directly by an adversary or they could be compromised nodes in the network. Such kind of attack may pose a very serious threat to distributed storage, routing protocols, data aggregation, voting, fair resource allocation, misbehavior detection, etc.
- *Wormhole attack:* Wormhole attack is an attack [31], where an adversary can tunnel messages received in one part of the network over a low latency link and replay them in a different part of the network. This attack may distort the network topology by making two distant nodes believe that they are neighbors and hence, it becomes a very serious threat to routing protocols.
- *Node replication attack:* In such attack [52], an adversary can intentionally put many replicas of a compromised node at many places in the network in order to incur inconsistency. Like the sybil attack, this attack can also render adversary the ability to subvert data aggression, misbehavior detection and voting protocols by injecting false data or suppressing legitimate data. Thus, in this attack an adversary can capture a set of sensor nodes in the network and then fabricate many replicas of those nodes with the information gathered from those captured nodes, and then place these replicas back into the strategic positions in the network for further malicious activities.
- *False reports injection attack:* Any compromised node can easily inject false data reports of non-existing events. When these fabricated reports are delivered to the base station then it can produce false alarms, waste valuable network resources, such as energy and bandwidth.
- *Man-in-the-middle attack:* In man-in-the-middle attack, an attacker has the ability to intercept messages in a public channel and then retransmits them by deleting or modifying the messages, so that the two original parties still believe as they are communicating with each other.
- *Node capture attack:* Capture of a certain number of nodes by the adversaries reveals the secret data stored in the nodes to the attacker. The attacker can then use those captured information to compromise secure communication among other non-compromised nodes.

We list the following essential security and functionality requirements for evaluating an ideal access control scheme designed for wireless sensor networks:

Security requirements

- *SR1. Withstand false reports injection attacks:* An attacker may try to inject false reports into the sensor networks. An access control protocol must prevent external parties from injecting reports into the existing sensor networks.
- *SR2. Withstand man-in-the-middle attacks:* An access control protocol must protect the man-in-the-middle attack from an adversary.
- *SR3. Resilience against node capture attacks:* The resilience against node capture attack of an access control scheme is measured by estimating the fraction of total secure communications that are compromised by a capture of c sensor nodes *not including* the communication in which the compromised nodes are directly involved. In other words, we wish to find out the effect of c sensor nodes being compromised on the rest of the network. For example, for any two non-compromised sensor nodes u and v, we need to find out the probability that the adversary can decrypt the secret communications between u and v when c sensor nodes are already compromised. An access control scheme must be highly resilient against node capture attacks.
- *SR4. Resilience against new node deployment attacks:* An access control scheme must defend against malicious node deployment attack, sybil attack, node replication attack and wormhole attack.

Functionality requirements

- *FR1. Dynamic node addition:* An access control scheme must allow nodes to dynamically join into the existing sensor network after initial deployment of nodes in order to replace malicious nodes or power-exhausted nodes.

- *FR2. Mutual authentication:* An access control scheme must provide mutual authentication between any two neighbor sensor nodes in order to verify mutually whether they are legitimate or not and if they are authenticated successfully they must establish pairwise key for future secure communication.
- *FR3. Network connectivity:* An access control scheme must provide very high network connectivity in the network, that is, any two neighbor nodes should be able to establish secret pairwise keys between them for future secure communication.
- *FR4. Communication overhead:* An access control scheme should be designed in such a way that it requires minimum number of message/packet transmissions during the authentication and key establishment phase in order to make it appropriate for practical applications.
- *FR5. Computational overhead:* An access control scheme should be computationally efficient.
- *FR6. Storage overhead:* An access control scheme should be such that the minimum information to be preloaded in the memory of sensor nodes before their deployment in the network for authentication and key establishment as well as for supporting dynamic nodes addition. Thus, the storage requirement in each sensor node must be minimum.
- *FR7. Scalability:* The designed access control scheme must be scalable, that is, it must support a large-scale network without involving the base station for authentication and key establishment purpose between neighbor nodes.

IV. Threat Model

For evaluating the security analysis and performance analysis of existing access control schemes, we use the following threat model as follows. In most applications, sensor networks operate in the hostile environments. We assume that sensor nodes can be physically captured by an attacker. Sensor nodes are not equipped with tamper-resistant hardware due to cost constraints and as a result, once a node is captured by an attacker, all the sensitive data as well as cryptographic information stored in its memory are revealed to the attacker. However, we assume that in any case, the base station (BS) will not be compromised by an attacker. We further assume that an attacker can directly deploy malicious nodes in the deployment field after the initial deployment of nodes.
As in [19], we make use of the Dolev-Yao threat model [22] in which two communicating parties (nodes) communicate over an insecure public channel. We adopt the similar threat model for WSNs where the channel is insecure and the endpoints (sensor nodes) cannot in general be trustworthy. Finally, we assume that an attacker can eavesdrop on all traffic, inject packets and reply old messages previously delivered.

V. Taxonomy of Security Protocols in WSNs

In this section, we give a taxonomy of security protocols in wireless sensor networks. The key management, access control, user authentication and user access control are the important security issues in wireless sensor networks. Figure 3 shows a taxonomy of security protocols in WSNs.
According to the probability of key sharing between a pair of sensor nodes, the key management schemes in WSNs can be divided into probabilistic and deterministic schemes. Pietro et al. [53] proposed a deterministic key management protocol based on the Logical Key Hierarchy (LKH). In this scheme, the base station is treated as a KDC (key distribution center) and all keys are logically distributed in a tree rooted at the base station. Zhu et al. [73] proposed a deterministic key management protocol called the Localized Encryption and Authentication Protocol (LEAP) for sensor networks. Lai et al. [42] proposed another deterministic scheme in which pairwise session keys between every two neighboring nodes are established. Eschenauer and Gligor [26] first introduced a probabilistic key pre-distribution scheme which relies on probabilistic key sharing among the nodes of a random graph. Chan et al. [8] proposed the q-composite keying scheme, where at least q common keys should be shared between the key rings of any two neighbor nodes in order to build a secure link between them. Liu and Ning [45] proposed a polynomial pool-based key pre-distribution scheme. Du et al. [24] presented another pairwise probabilistic key pre-distribution scheme which is similar to [45].
User authentication is a primary concern in a resource-constrained wireless sensor network before accessing real-time data from the nodes inside WSN. The real-time data can be given access directly to the external parties (users) those who are authorized to access data as and when they demand. User authentication is thus a very important primitive for providing access to real-time data inside WSN. The existing user authentications protocols proposed for wireless sensor networks usually fall into two categories: password-based user authentication and biometric-based user authentication. According to the authentication type and different factors used the protocols can be further divided into two categories: single factor and two factor authentication schemes. [65], [67], [59], [44], [39] and [46] are examples of single factor password based authentication schemes. [19], [51], [34], [37] and [17] are examples of two factor password-based authentication schemes. [70] is an example of biometric-based user authentication schemes.
Watro et al. [65] proposed a user authentication called TinyPK, which uses RSA algorithm [54] and Diffie-Hellman protocol [21]. However, there is a security flaw in TinyPK [19]. On receiving the user's public key, an attacker can encrypt a session key along with other parameters and send the encrypted message to the user. After receiving the encrypted message, the user easily believes that the message has come from the authorized sensor node. The user then decrypts the receiving encrypted message using his/her private key and also uses the session key for subsequent operations the attacker intends to perform. Wong et al. [67] proposed an efficient user authentication scheme which is based on user's password. It uses the efficient hash function. However, their scheme also is vulnerable to many logged in users with the same login-id threat, where a valid user's password can easily login to the sensor network. Further, their protocol suffers from stolen-verifier attack because both the GW-node (base station) as

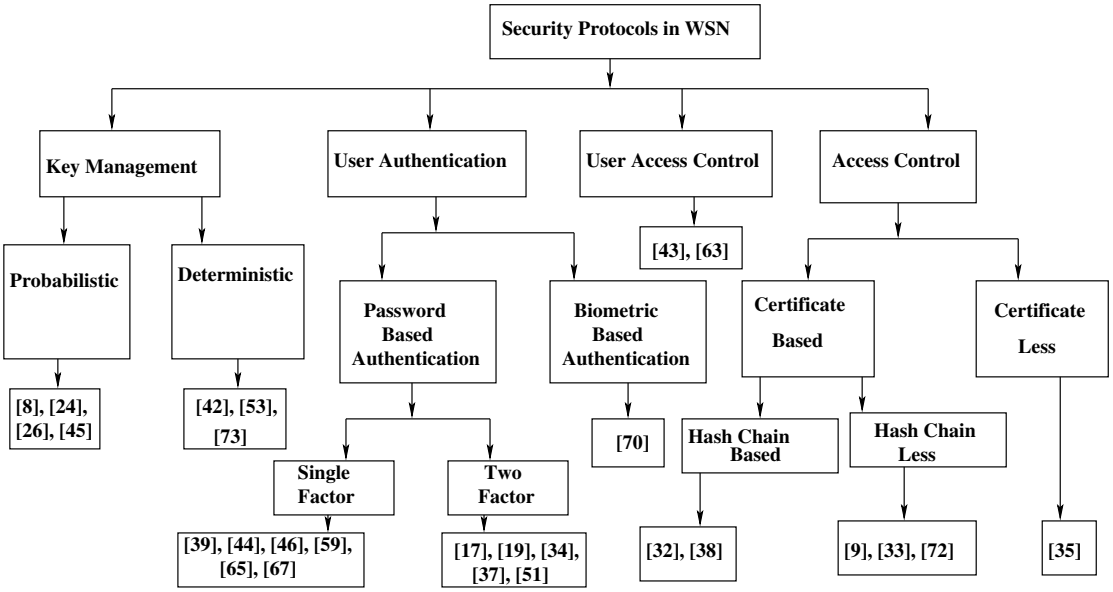

Figure. 3: Security protocols in WSNs: A taxonomy

well as login-node need to maintain the lookup table of registered users' credentials. M. L. Das [19] proposed a scheme to eliminate the flaws of Wong et al.'s scheme. However, it cannot resist denial-of-service attack and node compromise attack. Some improvements [37], [51] on [19] are proposed to withstand security flaws found in [19]. He et al. further proposed an enhanced scheme [30] based on M. L. Das's scheme [19]. Their scheme can withstand the security weaknesses such as vulnerabilities to an insider attack and to an impersonation attack. Vaidya et al. [60] further showed that M. L. Das's scheme [19] and Khan-Alghathbar's scheme [37] have security flaws and they remain vulnerable to various attacks including stolen smart card attacks. To overcome such security weaknesses of both schemes [19] and [37], an improved two-factor user authentication was proposed which is resilient to stolen smart card attacks and other common type of attacks. Fan et al. proposed a user authentication scheme [27], which is efficient and Denial-of-Service (DoS) resistant user authentication scheme for two-tiered WSNs. Chen and Shih [10] later pointed out that M. L. Das's scheme [19] fails to achieve mutual authentication. To overcome such problem, they proposed a robust mutual authentication protocol. Yuan et al.'s scheme [70] has better security as compared to that for M. L. Das's scheme [19] because the former scheme uses biometric verification along with the password verification of the user. Das et al. proposed recently a dynamic password-based user authentication scheme for hierarchical wireless sensor networks [17]. Their scheme is secure against different attacks and is better than existing user authentication schemes [19], [67], [19], [10], [30], [60], [27]. Yuan et al.'s scheme [70] is a biometric-based user authentication scheme which uses similar concept of [19] and it has same drawbacks as in M. L. Das's scheme [19]. It cannot still resist denial-of-service attack and node compromise attack.

To provide the access right to the legitimate users for different services in terms of information and resources of sensor network, user access control is very essential. Wang et al. [63] proposed a distributed user access control scheme under a realistic adversary model in which sensors can be compromised and user may collude. Le et al. [43] proposed another user access control scheme for wireless sensor networks based on public-key cryptography using ECC.

In this paper, we only concentrate our survey in the area of access control. Depending on the authentication type, the access control protocols are divided into two broad categories: certificate-based and certificate-less. Huang and Liu [35] proposed a certificate-less access control protocol based on one-way hash function. Certificate-based protocols can be further subdivided into hash-chain based and hash-chain less protocols. Huang [32] proposed an ECC-based access control using cascade hash chain. To overcome the limitations of [32], Kim and Lee [38] proposed an enhancement over [32] which additionally includes a renewal of hash chain phase. Zhou et al. [72] proposed a certificate-based ECC authentication using bootstrapping time and length of bootstrapping used in authentication. Huang [33] also proposed a dynamic access control scheme based on Schnorr signature and expiration time of the sensor nodes. Recently, Chatterjee et al. [9] has shown that Huang's scheme [33] is insecure against active attack and they also proposed an enhanced access control scheme based on ECC to withstand security flaw in [33]. Table 1 shows the recently proposed different access control schemes in WSNs. Depending on the authentication type, properties, and when published in the literature, we have described the schemes in this table.

VI. Review and Cryptanalysis of Existing Access Control Schemes in WSNs

In this section, we discuss the access control schemes proposed up-to date for wireless sensor networks in the literature. We review the schemes and their security weaknesses in this section. We use the notations shown in Table 2 for describing different access control schemes.

Random nonce is a one-time random bit-string which is usually used to achieve freshness. The public key of the certi-

Table 1: Brief overview of the access control schemes in WSN.

Scheme	Authentication type	Properties	Year of publication
Zhou et al. [72]	ECC based authentication	Bootstrapping time and length of bootstrapping used in authentication	2007
Huang-Liu [35]	Dynamic access control	Hash functions and XOR operations used in authentication	2008
Huang [32]	ECC based authentication	Cascade hash chain used in authentication	2009
Kim-Lee [38]	Enhancement over [32]	Mutual authentication with the base station and adds a renewal of hash chain phase	2009
Huang [33]	ECC and Schnorr signature based authentication	Expiration time used in authentication	2011
Chatterjee et al. [9]	Enhancement over [33]	Bootstrapping time, deployment version, and latest version checked variable used in authentication	2012

Table 2: Notations used in reviewing different access control schemes.

Symbol	Description
SN_u	Identifier of node SN_u
$E_q(a,b)$	An elliptic curve over finite field $GF(q)$
G	A base point on $E_q(a,b)$
k	Private key of CA (only known to CA)
Q	$Q = kG$, public key of CA
$K_{u,v}$	Symmetric secret key shared between nodes SN_u and SN_v
$H(\cdot)$	Secure one-way hash function
RN_u	Random nonce generated by node SN_u
$SN_u \rightarrow SN_v : M$	Message M sent from node SN_u to node SN_v
$A\|\|B$	Data A concatenates with data B
$EP_K(M)$	Public key encryption of message M using the key K
$DP_K(M)$	Public key decryption of message M using the key K

fication authority (CA) is $Q = kG$, where $kG = G + G + \ldots + G(k\,times) = \mathcal{O}$ is called the elliptic curve scalar multiplication in $E_q(a,b)$, $\mathcal{O}$ the point at infinity or zero point [40]. We may use the hash function $H(\cdot)$ as SHA-1 [3] or Quark [5].

A. Review of the Zhou et al.'s Access Control Scheme [72]

1) Description of the Protocol

Zhou et al. in [72] proposed an access control scheme based on elliptic curve cryptographic techniques for sensor network, which is more efficient than the schemes based on RSA. Their scheme consists of the following phases.

a. Pre-deployment phase
In pre-deployment phase, before a sensor network is deployed, the certification authority (CA) chooses a set of network parameters and preloads a set of node parameters as follows. CA chooses a set of network parameters which include: a finite field $GF(q)$, where q is a large odd prime of at least 160 bits for providing sufficient security in ECC; an elliptic curve $E_q(a,b)$ such that $y^2 = x^3 + ax + b \,(\mathrm{mod}\, q)$ with $4a^3 + 27b^2 \neq 0 \,(\mathrm{mod}\, q)$; a base point G of order n of at least 160 bits with $n > 4\sqrt{q}$; the CA's private key $k \in Z_n^* = \{1, 2, \ldots, n-1\}$; the CA's public key $Q = kG$. Once these node parameters are selected, CA preloads a set of node parameters to each sensor node SN_i which include: $E_q(a,b)$; G; Q; the bootstrapping time T_i when node SN_i bootstraps itself to join the network; the length of bootstrapping phase L_i during which the node SN_i is allowed to join the network; hash function $H(\cdot)$; node SN_i's private key s_i and its public key $P_i = s_iG = (x_{p_i}, y_{p_i})$, and the signature $\langle C_i, c_i \rangle$. The signature is crated using the elliptic curve digital signature algorithm (ECDSA) [36] for each node SN_i as follows. The CA chooses a random number k_i for each node SN_i and calculates $C_i = k_iG = (x_{c_i}, y_{c_i})$, where $c_i = {k_i}^{-1}(H(SN_i||T_i||L_i||P_i) + k.x_{c_i}) \,(\mathrm{mod}\, n)$.

b. Node deployment phase
In node deployment phase, sensor nodes bootstrap themselves and then start to establish communications among them. In each new node deployment, new sensor nodes are given a preset bootstrapping time different from that of the previously deployed nodes. It is assumed that the nodes are deployed in groups. Thus, nodes in one group have the same bootstrapping time and the length of the bootstrapping phase. However, these values for different groups may be different during deployment phases.

c. Authentication and key establishment phase
In node authentication phase, every new node broadcasts a message to inform its neighbors of its existence. In this phase, there are two kind of handshakes between nodes: one is the handshake between new nodes, and the other is the handshake between a new node and an old node. The purpose of these handshakes is to authenticate each node with its neighbor nodes as well as to establish secret keys between neighbor nodes.
After deployment, each new node SN_i needs to broadcast a message $\langle SN_i, T_i, L_i, P_i, C_i, c_i \rangle$ to its neighbors. Suppose a neighbor node SN_j hears this message from SN_i. Similarly, SN_i also hears a message $\langle SN_j, T_j, L_j, P_j, C_j, c_j \rangle$ from its neighbor SN_j. The handshaking between two neighbors is based on checking the validity of the bootstrapping time. SN_i first compares T_j of node SN_j with its own T_i. If $T_j \geq T_i$, then SN_j is considered as a new node. $T_j = T_i$ means that both nodes SN_i and SN_j are new nodes. SN_i further verifies whether SN_j is a new one by comparing T_j with its current time t and thus, if $|T_j - t| > L_j$, SN_i simple drops the message and considers SN_j as illegal node. If it is valid, then SN_i verifies SN_j's identity by verifying the signature (C_j, c_j) present in the message using ECDSA verification algorithm. If it is successful, SN_i confirms that SN_j is a valid new node deployed in the network and computes a

secret shared key with SN_j as $K_{i,j} = s_iP_j = s_is_jG$. In a similar fashion, node SN_j also verifies the identity of SN_i by verifying the signature (C_i, c_i) present in the message using ECDSA verification algorithm and calculates the shared secret key with SN_i as $K_{i,j} = s_jP_i = s_is_jG$. To confirm that both nodes share the same secret key $K_{i,j}$, they make use of the challenge-response protocol shown in Table 3 using the random nonces generated by each node as follows.

Table 3: Handshake between two new nodes in Zhou et al.'s scheme.

Node SN_i	Node SN_j
$\langle SN_i, T_i, L_i, P_i, C_i, c_i\rangle \rightarrow$	
	$\leftarrow \langle SN_j, T_j, L_j, P_j, C_j, c_j\rangle$
$\langle SN_j, SN_i, EP_{K_{i,j}}[RN_i]\rangle \rightarrow$	
	$\leftarrow \langle SN_i, SN_j, RN_i, EP_{K_{i,j}}[RN_j]\rangle$
$\langle SN_j, SN_i, RN_j\rangle \rightarrow$	

SN_i sends a message $\langle SN_j, SN_i, EP_{K_{i,j}}[RN_i]\rangle$ to SN_j by generating a random nonce RN_i. SN_j then decrypts the encrypted nonce to retrieve RN_i as $RN_i = DP_{K_{i,j}}[EP_{K_{i,j}}[RN_i]]$ and generates a random nonce RN_j. SN_j sends an acknowledgment $\langle SN_i, SN_j, RN_i, EP_{K_{i,j}}[RN_j]\rangle$ to SN_i. After receiving the acknowledgment, SN_i compares its own previously generated nonce with the received nonce RN_i in the message. If it is valid, SN_i further decrypts the encrypted nonce to get RN_j as $RN_j = DP_{K_{i,j}}[EP_{K_{i,j}}[RN_j]]$ and sends an acknowledgment $\langle SN_j, SN_i, RN_j\rangle$ to SN_j. Finally, SN_j verifies whether the received nonce RN_j matches with its own previously generated nonce. If it matches, SN_j considers SN_i as a legitimate node.

d. Dynamic node addition phase
If a new node is deployed after initial deployment, the base station preloads with necessary information in its memory prior to its deployment using the above pre-deployment phase. Suppose a newly deployed node SN_i wants to authenticate and establish secret key with its old neighbor node SN_j. Consider the following handshake between the new node SN_i and the old node SN_j. When SN_j hears the broadcasted message $\langle SN_i, T_i, L_i, P_i, C_i, c_i\rangle$ from SN_i, it checks the validity of T_i and also verifies SN_i's identity by verifying the signature (C_i, c_i) present in the message using ECDSA verification algorithm and calculates the shared secret key with SN_i as $K_{i,j} = s_jP_i = s_is_jG$. In response, SN_j sends a reply message $\langle SN_i, SN_j, T_j, L_j, P_j, C_j, c_j, EP_{K_{i,j}}[RN_j]\rangle$ to SN_i. After receiving the message, SN_i verifies SN_j's identity by verifying the signature (C_j, c_j) present in the message using ECDSA verification algorithm and calculates the shared secret key with SN_j as $K_{i,j} = s_iP_j = s_is_jG$. SN_i further decrypts encrypted nonce using the computed key $K_{i,j}$ to get RN_j as $RN_j = DP_{K_{i,j}}[EP_{K_{i,j}}[RN_j]]$. SN_i then sends the message $\langle SN_j, SN_i, RN_j, EP_{K_{i,j}}[RN_i]\rangle$ to SN_j. After receiving the message from SN_i, SN_j decrypts the encrypted nonce using the key $K_{i,j}$ as $RN_i = DP_{K_{i,j}}[EP_{K_{i,j}}[RN_i]]$ and sends an acknowledgment $\langle SN_i, SN_j, RN_i\rangle$ to SN_i. SN_i then verifies whether the received nonce matches with its previously generated nonce RN_i. If it matches, SN_i also confirms that SN_j a new legitimate node deployed in the network. Note that SN_i does not need to check the validity of T_j since SN_j is not a new node. The flow of messages in this phase is summarized in Table 4.

Table 4: Handshake between a new node and old node in Zhou et al.'s scheme.

Node SN_i	Node SN_j
$\langle SN_i, T_i, L_i, P_i, C_i, c_i\rangle \rightarrow$	
	$\leftarrow \langle SN_i, SN_j, T_j, L_j, P_j, C_j,$ $c_j, EP_{K_{i,j}}[RN_j]\rangle$
$\langle SN_j, SN_i, RN_j, EP_{K_{i,j}}[RN_i]\rangle \rightarrow$	
	$\leftarrow \langle SN_j, SN_i, RN_i\rangle$

2) Cryptanalysis of the Protocol

The Zhou et al.'s scheme assumes that each sensor node can sustain a tolerance time interval before it is compromised. However, if a sensor node is captured during its bootstrapping phase, all the secret information stored in its memory are also compromised. Using those information, an attacker can launch other attacks. Though the node bootstrapping phase is usually very short, if an attacker compromises a sensor node during that period then Zhou et al.'s scheme is insecure against other attack such as a node replication attack. Due to this problem, this scheme may not be convenient for some practical applications. Further, this scheme is not secure against node capture attack.

B. Review of the Huang-Liu's Scheme [35]

1) Description of the Protocol

Huang and Liu [35] proposed an energy efficient and low computational overhead dynamic access control protocol in WSN using hash functions and XOR operations.

a. Pre-deployment phase
It is assumed that there is a number of r neighborhood nodes in a designated area. In pre-deployment phase, the base station generates different secret keys for all the sensor nodes deployed in that area of the target field. The base station preloads the secret key k_i, the one-way hash function $H(\cdot)$ and the node identity SN_i to each node SN_i's memory prior to its deployment.

b. Node deployment phase
After the deployment of sensor nodes in the target field, the base station generates the pairwise secret keys $SK_{i,j}$ by computing the XOR of the hash values of one node's identity with other node's secret key for each pair of nodes SN_i and SN_j in the sensor network, where $SK_{i,j} = H(k_i||SN_j) \oplus H(k_j||SN_i)$, $(i = 1, 2, \ldots, r; j = i+1, i+2, \ldots, r)$. Once these pairwise keys are generated, the base station broadcasts all the pairwise secret keys $SK_{i,j}$ to the nodes in the network.

c. Authentication and key establishment phase
In authentication and key establishment process, any two nodes SN_i and SN_j can authenticate to each other and establish a secret key as follows. In this phase, any node SN_i first

computes the hash value of its own secret key k_i with other node SN_j's identity as $H(k_i||SN_j)$. After that SN_i computes $a_i = SK_{i,j} \oplus H(k_i||SN_j) = H(k_j||SN_i)$. Then SN_i generates a random number t_i and computes the hash value z_i as $z_i = H(a_i||t_i) = H(H(k_j||SN_i)||t_i)$. Node SN_i then sends the message $\langle SN_i, z_i, t_i\rangle$ to the node SN_j.

After receiving the message from node SN_i, node SN_j verifies the value of z_i by verifying the condition $z_i = H(H(k_j||SN_i)||t_i)$ and if it passes, node SN_j considers that node SN_i is a legitimate node. SN_j then generates a_j as $a_j = SK_{i,j} \oplus H(k_j||SN_i) = H(k_i||SN_j)$ using the broadcasted information $SK_{i,j}$ and its own secret key k_j. SN_j also generates a random number t_j and computes z_j as $z_j = H(a_j||t_j) = H(H(k_i||SN_j)||t_j)$ and finally sends the message $\langle SN_j, z_j, t_j\rangle$ to node SN_i for authentication. When SN_i receives the message from SN_j, SN_i verifies the value of z_j by verifying the condition $z_j = H(H(k_i, SN_j)||t_j)$ and if the validation is successful, SN_i considers that SN_j is also a legitimate node. SN_i then computes a shared session key $K_{i,j} = H((H(k_j||SN_i) \oplus t_i) \oplus (H(k_i||SN_j) \oplus t_j)) = H((a_i \oplus t_i)||(a_j \oplus t_j))$ and computes another hash value $y_{ij} = H(K_{i,j}||(t_i \oplus t_j))$ and delivers y_{ij} to node SN_j for establishing mutual authentication purpose.

Once SN_j receives the message from SN_i, SN_j computes the same secret key shared with SN_i as $K_{i,j} = H((H(k_j||SN_i) \oplus t_i) \oplus (H(k_i||SN_j) \oplus t_j)) = H((a_i \oplus t_i)||(a_j \oplus t_j))$ and checks if $H(K_{i,j}||(t_i \oplus t_j)) = y_{ij}$ holds or not. If it holds, the connection is established between these two nodes and they use the common secret key $K_{i,j}$ for their future secret communication.

d. Dynamic node addition phase

Suppose a new node with identity SN_{r+1} be deployed in the network. Then the base station preloads the randomly generated secret key k_{r+1}, the one-way hash function $H(\cdot)$ and the node identity SN_{r+1} in its memory. After its deployment, the base station only computes the secret keys $SK_{i,r+1}$ as $SK_{i,r+1} = H(k_i||SN_{r+1}) \oplus H(k_{r+1}||SN_i)$, $i = 1, 2, \ldots, r$ and broadcasts these information to the existing nodes in the network. Then the deployed node SN_{r+1} authenticates and establishes keys with other nodes in a same manner as in the authentication and key establishment phase. The transmission of messages during the authentication and key establishment phase is summarized in Table 5.

Table 5: Authentication and key establishment phase in Huang-Liu's scheme.

Node SN_i	Node SN_j
$\langle SN_i, z_i, t_i\rangle \rightarrow$	
	$\leftarrow \langle SN_j, z_j, t_j\rangle$
$\langle SN_i, y_{ij}\rangle \rightarrow$	

2) Cryptanalysis of the Protocol

According to the threat model (given in Section IV), if an adversary captures a sensor node, say SN_u, the adversary will get the secret key k_u and its identity from its memory. Since all the secret keys $SK_{i,j}$'s for each pair of nodes are broadcasted by the base station, the adversary also knows these secret keys $SK_{i,j}$. We propose the following node replication attack in this scheme as follows. The adversary preloads the extracted information k_u and identity SN_u along with the hash function $H(\cdot)$ in the memory of a fake sensor node FSN_u and deploys in some other part of the network. After deployment, the fake deployed sensor node FSN_u will be successful to authenticate with its neighbor nodes and establish secret keys with them. For example, in order to authenticate and establish secret shared key with a neighbor node SN_v, FSN_u first computes $a_u = SK_{u,v} \oplus H(k_u||SN_v) = H(k_v||SN_v)$ using the broadcasted information $SK_{u,v}$. Then FSN_u generates a random number t_u and computes the hash value z_u as $z_u = H(a_u||t_u) = H(H(k_v||SN_u)||t_u)$. Node FSN_u then sends the message $\langle SN_u, z_u, t_u\rangle$ to the node SN_v. After receiving the message, node SN_v checks the authenticity of node FSN_u by checking z_u. In a similar way, SN_v generates a_v as $a_v = SK_{u,v} \oplus H(k_v||SN_u) = H(k_u||SN_v)$ using the previous broadcasted information $SK_{u,v}$ and its own secret key k_v. SN_v then generates a random number t_v and computes $z_v = H(H(k_u||SN_v)||t_v)$ and sends the message $\langle SN_v, z_v, t_v\rangle$ to node FSN_u. FSN_u checks the validity of node SN_v and establishes a shared secret key with SN_v as $K_{u,v} = H((H(k_v||SN_u) \oplus t_u) \oplus (H(k_u||SN_v) \oplus t_v))$ and delivers $y_{uv} = H(K_{u,v}||(t_u \oplus t_v))$ to node SN_v for establishing mutual authentication. Node SN_v also establishes the same secret key $K_{u,v}$ shared with SN_u. Thus, the Huang-Liu's scheme is insecure against node replication attacks.

The Huang-Liu's scheme can resist other attacks such as sybil attack, wormhole attack, man-in-the-middle attack, and false reports injection attack. However, this scheme cannot resist node capture attack.

C. Review of the Huang's Scheme [32]

1) Description of the Protocol

Huang [32] proposed an access control scheme whose authentication based on elliptic curve cryptography and hash chain.

a. Pre-deployment phase

In this protocol, it is assumed that there is a number of r neighborhood nodes with identities $\{SN_1, SN_2, \ldots, SN_r\}$ in a designated area. The base station generates r secret keys k_i for each node SN_i, $(i = 1, 2, \ldots, r)$, a base point G of order n (n is at least 160 bits) of the elliptic curve $E_q(a, b)$ and selects a one-way hash function $H(\cdot)$. The base station then preloads the secret key k_i, the elliptic curve $E_q(a, b)$, the base point G, n and the hash function $H(\cdot)$ and the identity SN_i to each node SN_i.

b. Node deployment phase

After deployment of nodes in the designated area, the base computes the hash values $H^z(k_i)$ $(i = 1, 2, \ldots, r)$ and broadcasts all these computed hash values $H^z(k_i)$ along with z to the group of nodes $\{SN_1, SN_2, \ldots, SN_r\}$. Here z is considered a large constant number and it may be considered as a limitation of the nodes. The expression $H^l(k)$ is the application of l cascade hash operations starting from k, that is, $H^l(k) = H(H^{l-1}(k))$. Note that it is computationally easy to compute the hash value $H^l(k)$

given $H^{l-1}(k)$. However, it is computationally infeasible problem to compute the hash value $H^{l-1}(k)$ given $H^{l}(k)$ due to one-way property of the hash function $H(\cdot)$.

c. Authentication and key establishment phase
Let two neighbor nodes SN_i and SN_j wish to authenticate and establish secret key between them. Assume that the nodes SN_i and SN_j have passed through authentication u times and v times, respectively. Note that the broadcasting hash chain for SN_i and SN_j are then $H^{z-u}(k_i)$ and $H^{z-v}(k_j)$, respectively.
Node SN_i computes a point over the elliptic curve $E_q(a,b)$ as $A_i = t_iG = (A_{x_i}, A_{y_i})$ by generating a random number t_i $(< n)$. Also it computes a hash value $s_i = H(A_{x_i} ||H^{z-u-1}(k_i))$. Then for authentication it sends the message $\langle SN_i, A_i, s_i\rangle$ to node SN_j. Similarly, node SN_j also generates a random number t_j $(< n)$, computes $A_j = t_jG = (A_{x_j}, A_{y_j})$, $s_j = H(A_{x_j}||H^{z-v-1}(k_j))$ and sends the message $\langle SN_j, A_j, s_j\rangle$ to node SN_i. After receiving the message from SN_j, SN_i computes the shared session key $K_{i,j} = t_iA_j = (K_{x_{ij}}, K_{y_{ij}})$ and the hash value $z_i = H(K_{x_{ij}}||H^{z-u-1}(k_i))$. SN_i then sends the message $\langle z_i, H^{z-u-1}(k_i)\rangle$ to node SN_j.
After receiving the message from node SN_i, node SN_j first computes the secret session key shared with SN_i as $K_{i,j} = t_jA_i = (K_{x_{ij}}, K_{y_{ij}})$ and then verifies whether $H(H^{z-u-1}(k_i)) = H^{z-u}(k_i)$, $H(K_{x_{ij}}||H^{z-u-1}(k_i)) = z_i$ and $H(A_{x_i}||H^{z-u-1}(k_i)) = s_i$ hold or not. If all the verifications hold, SN_j can make sure that SN_i is a legitimate node. SN_j further computes $z_j = H(K_{x_{ij}}||H^{z-v-1}(k_j))$ and sends the message $\langle z_j, H^{z-v-1}(k_j)\rangle$ to node SN_i. After receiving the message from SN_j, SN_i verifies whether $H(H^{z-v-1}(k_j)) = h^{z-v}(k_j)$, $H(A_{x_j}||H^{z-v-1}(k_j)) = s_j$ and $H(K_{x_{ij}}||H^{z-v-1}(k_j)) = z_j$ hold or not. If all these conditions are true, SN_i can also make sure that SN_j is a legitimate. As in Huang-Liu's scheme, Huang's scheme also achieves the mutual authentication between SN_i and SN_j. Nodes SN_i and SN_j finally update their broadcasting hash chain as $H^{z-u-1}(k_i)$ and $H^{z-v-1}(k_j)$, respectively and inform all the members of the group using the base station. This phase is summarized in Table 6.

Table 6: Authentication and key establishment phase in Huang's scheme [32].

Node SN_i	Node SN_j
$\langle SN_i, A_i, s_i\rangle \xrightarrow{}$	
	$\xleftarrow{} \langle SN_j, A_j, s_j\rangle$
$\langle z_i, H^{z-u-1}(k_i)\rangle \xrightarrow{}$	
	$\xleftarrow{} \langle z_j, H^{z-v-1}(k_j)\rangle$

d. Dynamic node addition phase
Suppose a new node SN_{r+1} be deployed in the existing network. For this purpose, the base station generates a secret key k_{r+1} randomly and then preloads k_{r+1}, the hash function $H(\cdot)$ and its identity in the memory of SN_{r+1}. After its deployment, the base needs only to compute $H^z(k_{r+1})$ and broadcast this information to all the existing nodes in the network. Then the new node SN_{r+1} authenticates and establishes secret keys with its neighbor nodes using the same authentication and key establishment phase described above.

2) Cryptanalysis of the Protocol

Kim and Lee [38] pointed out that Huang's scheme [32] has several security weaknesses such as it is insecure against the replay attack and the new node masquerading attack in presence of an active adversary due to absence of authentication procedure for the base station. Further, Kim and Lee show that Huang's scheme has the lack of renewability for the hash chain. We discuss these weaknesses as follows.

- *Replay attack:* The Huang's scheme only allows the unilateral authentication, that is, only the base station authenticates the sensor nodes in the network. Suppose an attacker wants to perform the replay attack against a node SN_i and that node has already passed through authentication u times. Then the attacker has collected a group of secret values $\{H^{z-u}(k_i), H^{z-u-1}(k_i), \ldots, H^{z}(k_i)\}$ by intercepting the secret change messages from the base station. Now if the attacker broadcasts a message to all the nodes in the network except to the node SN_i and the base station with the same information from the base station after changing a secret value with $H^{z-u-r}(k_i)$ from the collected group, where $0 < r < u$. Since there is no need for authenticity of the message from the base station, the existing nodes believe that the message is valid and it has come from the base station legally. Hence, all the existing nodes in the group except for SN_i will use the secret value $H^{z-u-r}(k_i)$ in the authentication and key establishment phase. However, the node SN_i still use the secret $H^{z-u+1}(k_i)$ for the authentication and key establishment phase and it will be rejected by all the other nodes. Thus, Huang's scheme is insecure against replay attack.

- *New node masquerading attack:* When a new node SN_{r+1} needs to be deployed in the network, the base station preloads a secret key k_{r+1}, the hash function $H(\cdot)$ and its identity in the memory of SN_{r+1}. After that the base station computes $H^z(k_{r+1})$ and broadcast this information to all the existing nodes in the network as an authenticator of the new node SN_{r+1}. Suppose an attacker wants to deploy a fake node in the network and he/she wants to masquerade a new legal node. As pointed out by Kim and Lee, the attacker requires to select a random number k'_{r+1} as its secret key, compute a randomized hash chain $H^c(k'_{r+1})$, where c is a random value, and broadcast the chain to all the existing nodes except to the base station. Kim and Lee also noted that there is no need to synchronize c with z. After receiving the broadcast message from the attacker, all the existing sensor nodes believe that the attacker is a legal node because there is no way to check the authenticity of the message from the attacker in Huang's scheme. As a result, the attacker can easily communicate with other nodes by using the secret value from the hash chain $H^c(k'_{r+1})$. Note that the authentication and key establishment phase of any existing sensor node in the network with the attacker's fake node remains same as Huang's original authentication and key establishment phase. In this way, Huang's scheme is insecure against new node masquerading attack.

- *Lack of hash chain renewability:* It is noted that the authenticity in Huang's scheme is only based on the hash chain $H^z(k_i)$ for each sensor node SN_i. Thereby, it would be possible that some nodes which have no left over the secret value in the hash chain. This means that those nodes has already performed $z-1$ times of the authentication and key establishment phase. Hence, Huang's scheme has also the lack of hash chain renewability.

In addition, the Huang's scheme cannot resist false reports injection attack and node capture attack.

D. Review of the Kim-Lee's Scheme [38]

1) Description of the Protocol

In order to remedy the security weaknesses found in Huang's scheme such as replay attack, new node masquerading attack and lack of hash chain renewability, Kim and Lee [38] proposed an enhancement over Huang's scheme. Various phases of Kim-Lee's scheme are discussed below.

a. Pre-deployment phase
As in Huang's scheme, it is assumed that there is a number of r neighborhood nodes with identities $\{SN_1, SN_2, \ldots, SN_r\}$ in a designated area. The base station generates its own secret key k_{bs}, a random number a_{bs}, selects a one-way hash function $H(\cdot)$ and computes its own hash chain $H^z(k_{bs}||a_{bs})$, where z is a random number.
The base station then generates r secret keys k_i for all r sensor nodes SN_i $(i = 1, 2, \ldots, r)$ and r random numbers a_i $(i = 1, 2, \ldots, r)$ for all r nodes. The base station finally preloads k_i, a_i, $H(\cdot)$, z and $H^z(k_{bs}||a_{bs})$ into the memory of each node SN_i. The base station also preloads ECC parameters: the elliptic curve $E_q(a, b)$, the base point G, the order of base point n into the memory of each node SN_i.

b. Node deployment phase
After deployment of nodes in the designated area, the base computes the hash values $H^z(k_i||a_i)$ $(i = 1, 2, \ldots, r)$ and broadcasts all these computed hash values $H^z(k_i||a_i)$ to the group of nodes $\{SN_1, SN_2, \ldots, SN_r\}$.

c. Authentication and key establishment phase
The authentication and key establishment phase of Kim-Lee's scheme is similar to that for Huang's scheme. Assume that two neighbor nodes SN_i and SN_j have passed through authentication u times and v times, respectively. It is also noted that the broadcasting hash chain for SN_i and SN_j are then $H^{z-u}(k_i||a_i)$ and $H^{z-v}(k_j||a_j)$, respectively.
In this phase, node SN_i first computes a point over the elliptic curve $E_q(a, b)$ as $A_i = t_iG = (A_{x_i}, A_{y_i})$ by generating a random number t_i $(< n)$ and then computes a hash value $s_i = H(A_{x_i}||H^{z-u-1}(k_i||a_i))$. It sends the message $\langle SN_i, A_i, s_i\rangle$ to node SN_j. Similarly, node SN_j generates a random number t_j $(< n)$, computes $A_j = t_jG = (A_{x_j}, A_{y_j})$, $s_j = H(A_{x_j}||H^{z-v-1}(k_j||a_j))$ and sends the message $\langle SN_j, A_j, s_j\rangle$ to node SN_i. After receiving the message from SN_j, node SN_i first computes the shared session key $K_{i,j} = t_iA_j = (K_{x_{ij}}, K_{y_{ij}})$ and the hash value $z_i = H(K_{x_{ij}}||H^{z-u-1}(k_i||a_i))$, and then sends the message $\langle z_i, H^{z-u-1}(k_i||a_i)\rangle$ to node SN_j. When the message from node SN_i is received by node SN_j, it first computes the same secret session key shared with SN_i as $K_{i,j} = t_jA_i = (K_{x_{ij}}, K_{y_{ij}})$. It then continues to verify whether the conditions $H(H^{z-u-1}(k_i||a_i)) = H^{z-u}(k_i||a_i)$, $H(K_{x_{ij}}||H^{z-u-1}(k_i||a_i)) = z_i$ and $H(A_{x_i}||H^{z-u-1}(k_i||a_i)) = s_i$ hold or not. If all these conditions hold good, SN_j can make sure that SN_i is a legitimate node. SN_j further computes $z_j = H(K_{x_{ij}}||H^{z-v-1}(k_j||a_j))$ and sends the message $\langle z_j, H^{z-v-1}(k_j||a_j)\rangle$ to node SN_i. After receiving the message from SN_j, SN_i further verifies whether the conditions $H(H^{z-v-1}(k_j||a_j)) = h^{z-v}(k_j||a_j)$, $H(K_{x_{ij}}||H^{z-v-1}(k_j)) = z_j$ and $H(A_{x_j}||H^{z-v-1}(k_j||a_j)) = s_j$ hold or not. If all these hold, SN_i can also make sure that SN_j is a legitimate. Thus, the mutual authentication between SN_i and SN_j is achieved. Nodes SN_i and SN_j finally update their broadcasting hash chain as $H^{z-u-1}(k_i)$ and $H^{z-v-1}(k_j)$, respectively and inform all the members of the group using the base station. This phase is summarized in Table 7.

Table 7: Authentication and key establishment phase in Kim-Lee's scheme.

Node SN_i	Node SN_j
$\langle SN_i, A_i, s_i\rangle \xrightarrow{}$	
	$\xleftarrow{\langle SN_j, A_j, s_j\rangle}$
$\langle z_i, H^{z-u-1}(k_i\|\|a_i)\rangle \xrightarrow{}$	
	$\xleftarrow{\langle z_j, H^{z-v-1}(k_j\|\|a_j)\rangle}$

d. Dynamic node addition phase
If a new node SN_{r+1} with identity SN_{r+1} is added in the network, the base station needs to generate a secret key k_{r+1}, a random number a_{r+1} and preloads these information along with $H(\cdot)$, ECC parameters: the elliptic curve $E_q(a, b)$, the base point G, the order of base point n, $H^{z-w}(k_{bs}||a_{bs})$, the already nodes' secret values $H^{z-u}(k_i||a_i)$ $(i = 1, 2, \ldots, r)$ into the memory of SN_{r+1}, assuming that the base station has already passed through new node addition or updating hash chain w times. It is also assumed that each node SN_i has passed through authentication u times.
The base station computes $H^z(k_{r+1}||a_{r+1})$ and $z_{bs} = H(H^z(k_{r+1}||a_{r+1})||H^{z-w-1}(k_{bs}||a_{bs}))$ and broadcasts the message $\langle SN_{r+1}, H^z(k_{r+1}||a_{r+1}), z_{bs}\rangle$ about the new node addition to all existing nodes in the network. The base station further broadcasts its secret value $H^{z-w-1}(k_{bs}||a_{bs})$ for authenticity check of the previous broadcast.
After receiving the broadcasted information, each node SN_i in the network checks whether the conditions $H^{z-w}(k_{bs}||a_{bs}) = H(H^{z-w-1}(k_{bs}||a_{bs}))$ and $H(H^z(k_{r+1}||a_{r+1})||H^{z-w-1}(k_{bs}||a_{bs})) = z_{bs}$ are satisfied or not. If these are valid, each node ensures that the new node SN_{r+1} is a legitimate node and also updates the broadcasting hash chain for the base station which be $H^{z-w-1}(k_{bs}||a_{bs})$. This phase is summarized in Table 8.
e. Renewal of hash chain phase
This phase is similar to that of new node addition phase. If a node SN_i has only a secret value or shortage in hash chain, it requires to renew the hash chain. In order to renew hash chain, SN_i first creates a request message M_{req},

Table 8: New node addition phase in Kim-Lee's scheme.

Node SN_i	Base station
	$\langle SN_{r+1}, H^z(k_{r+1}\|\|a_{r+1}), z_{bs}\rangle$ $\longleftarrow$
	$\langle H^{z-w-1}(k_{bs}\|\|a_{bs})\rangle$ $\longleftarrow$

computes the value $z_i = H(M_{req}||H^{z-u-1}(k_i||a_i))$ and sends the message $\langle SN_i, M_{req}, z_i\rangle$ to the base station. After this it also broadcasts $H^{z-u-1}(k_i||a_i)$. The base station then verifies whether the conditions $H^{z-u}(k_i||a_i) = H(H^{z-u-1}(k_i||a_i))$ and $H(M_{req}||H^{z-u-1}(k_i||a_i)) = z_i$ are satisfied or not. If these hold, the base station believes the authenticity of node SN_i and then increments a_i as $a_i = a_i + 1$, computes the new hash chain $H^z(k_i||a_i)$, $z_{bs} = H(H^z(k_i||a_i)||H^{z-w-1}(k_{bs}||a_{bs}))$. After that the base station broadcasts $\langle SN_i, H^z(k_i||a_i), z_{bs}\rangle$ and $H^{z-w-1}(k_{bs}||a_{bs})$ to all existing nodes in the network. After receiving the broadcasted information, the node SN_i checks the conditions $H(H^{z-w-1}(k_{bs}||a_{bs})) = H^{z-w}(k_{bs}||a_{bs})$ and $H(H(k_i||a_i)||H^{z-w-1}(k_{bs}||a_{bs})) = z_{bs}$. If these conditions are valid, the node SN_i makes sure that the renewal message from the base station is legitimate. Then the node SN_i increases the random number a_i as $a_i = a_i + 1$ and updates its own broadcasted chain with $H^z(k_i||a_i)$. Similarly, other existing nodes also verify the validity of the broadcasted renewal message from the base station. This phase is also summarized in Table 9.

Table 9: Renewal of hash chain phase in the Kim-Lee's scheme.

Node SN_i	Base station
$\langle SN_i, M_{req}, z_i\rangle$ $\longrightarrow$	
$\langle H^{z-u-1}(k_i\|\|a_i)\rangle$ $\longrightarrow$	
	$\langle SN_i, H^z(k_i\|\|a_i), z_{bs}\rangle$ $\longleftarrow$
	$\langle H^{z-w-1}(k_{bs}\|\|a_{bs})\rangle$ $\longleftarrow$

2) Cryptanalysis of the Protocol

Though Kim-Lee's scheme is the enhancement over Huang's scheme [32], Zeng et al. in [71] demonstrates that Kim-Lee's scheme is vulnerable to a new node masquerading attack and a legal node masquerading attack. They showed that an attacker can easily intercept the secret values without being detected. The more details of these attacks are explained in [71].

Later, Shen et al. in [57] showed that Kim-Lee's scheme is vulnerable to active attacks in authentication and key establishment phase, new node addition phase and renewal of hash chain phase. During the authentication and key establishment phase, when a node SN_i sends the message $\langle SN_i, A_i, s_i\rangle$ to its neighboring node SN_j, the attacker can easily intercept A_i and $H^{z-u-1}(k_i||a_i)$. In such case, the adversary can block the correct values of A_i and $H^{z-u-1}(k_i||a_i)$ and resubmit the distorted values A_i' and $H^{z-u-1}(k_i||a_i)'$ after modifying the values of A_i and $H^{z-u-1}(k_i||a_i)$ to node SN_j. When the node SN_j verifies the verification conditions in this phase, all the equations will not hold. Similarly, when the node SN_j communicates with node SN_i, in a similar way the adversary will be able to intercept and modify the values of A_j and $H^{z-v-1}(k_j||a_j)$ to make the authentication unsuccessful. In the similar way, Shen et al. showed that Kim-Lee's scheme is also insecure to such active attacks during new node addition and renewal of hash chain phases. Thus, Kim-Lee's scheme is insecure against several attacks.

E. Review of the Huang's Scheme [33]

1) Description of the Protocol

Huang's scheme [33] is based on ECC and the concept of Schnorr signature [55]. This access control scheme uses the concept of time bound in which once time period has elapsed, the sensor nodes in wireless network cannot access any data for future time period. For that purpose, each node is given its own expiration time w. A node can achieve authentication and establishment of secret keys with other nodes in the time period z if and only if $z <= w$. Once the time period $z > w$ elapses, any node is not allowed for authentication and key establishment with other nodes. This scheme is discussed briefly as follows.

a. Pre-deployment phase

In this phase, the base station first selects a secret key x and computes its public key $Q = xG$ over the elliptic curve $E_q(a, b)$, where G is the base point whose order is n (of at least 160 bits).

As in [32], it is also assumed that there is a number of v neighborhood nodes with identities $\{SN_1, SN_2, \ldots, SN_v\}$ in a designated area. For each node SN_i, the base station generates a random number r_i, an expiration time $w_i (< t)$ and computes the public value $R_i = r_iG = (R_{x_i}, R_{y_i})$, the value $s_i = r_i + c_ix \pmod q$, where $c_i = H(SN_i||R_{x_i}||R_{y_i}||w_i)$. The base station preloads $E_q(a, b)$, Q, G, n, one-way hash function $H(\cdot)$, w_i, and (R_i, s_i) to each node SN_i's memory $(i = 1, 2, \ldots, r)$.

b. Authentication and key establishment phase

Let the time period be T. Suppose two nodes SN_i and SN_j want to authenticate and establish secret key between them. Node SN_i first generates a random number t_i $(< n)$, computes the public value $A_i = t_iG$ and sends the message $\langle SN_i, A_i\rangle$ to node SN_j. Similarly, node SN_j also generates a random number t_j $(< n)$, computes the public value $A_j = t_jG$ and sends the message $\langle SN_j, A_j\rangle$ to node SN_i. Node SN_i then computes the secret key shared with SN_j as $K_{i,j} = t_iA_j = (K_{x_{ij}}, K_{y_{ij}})$ and also computes the signature $z_i = t_i + e_is_i \pmod q$, where $e_i = H(SN_i||K_{x_{ij}}||K_{y_{ij}})$, and sends the message $\langle z_i, R_i, w_i\rangle$ to node SN_j. Similarly, node SN_j also computes the same secret key shared with SN_i as $K_{i,j} = t_jA_i = (K_{x_{ij}}, K_{y_{ij}})$. After that SN_j computes the signature $z_j = t_j + e_js_j \pmod q$, where $e_j = H(SN_j||K_{x_{ij}}||K_{y_{ij}})$, and sends the message $\langle z_j, R_j, w_j\rangle$ to node SN_i. Finally, nodes SN_i and SN_j verify the authenticity of each other using values of z_i, z_j, w_i and w_j as follows. SN_i checks whether $w_j > T$ and $z_jG = A_j + e_j(R_j + c_jQ)$, where $c_j = H(SN_j||R_{x_j}||R_{y_j}||w_j)$, $e_j = H(SN_j||K_{x_{ij}}||K_{y_{ij}})$ and $R_j = (R_{x_j}, R_{y_j})$. If these hold, SN_i makes sure that SN_j is a legitimate. Similarly, node SN_j checks whether $w_i > T$ and $z_iG = A_i +$

$e_i(R_i + c_iQ)$, where $c_i = H(SN_i||R_{x_i}||R_{y_i}||w_i)$, $e_i = H(SN_i||K_{x_{ij}}||K_{y_{ij}})$ and $R_i = (R_{x_i}, R_{y_i})$. If these hold, SN_j also makes sure that SN_i is a legitimate. The exchanges of messages in this phase are summarized in Table 10.

Table 10: Authentication and key establishment phase in Huang's scheme [33].

Node SN_i	Node SN_j
$\langle SN_i, A_i \rangle \xrightarrow{}$	
	$\xleftarrow{} \langle SN_j, A_j \rangle$
$\langle z_i, R_i, w_i \rangle \xrightarrow{}$	
	$\xleftarrow{} \langle z_j, R_j, w_j \rangle$

c. Dynamic node addition phase
The same procedure is applied for a new node N_{r+1} deployment. The base station needs to preload the information as done in pre-deployment phase for other nodes. After deployment, the new node performs the above authentication and key establishment phase in order to authenticate and establish pairwise

2) Cryptanalysis of the Protocol

Huang's scheme has fatal weaknesses such as it is insecure against active attacks (for example, man-in-the-middle attacks). In such an active attack, during the authentication and key establishment phase an adversary (attacker) can block the correct A_i and resubmit the distorted A_i' to node SN_j after modifying the value of A_i. Node SN_i will not be able to pass the authentication in node SN_j, and hence the node SN_j considers the node SN_i as an illegitimate node because the later verification equation cannot hold. On the other hand, when node SN_j communicates with node SN_i, the adversary can also intercept and modify the value of A_j, and resubmit the distorted value of A_j' and then the authentication of node SN_j again fails. This attack is described below [9]:
The attacker $\mathcal{A}$ first intercepts the message $\langle SN_i, A_i \rangle$ sent by node SN_i towards node SN_j and blocks this message. Note that it is a computationally hard problem to find t_i from A_i for the attacker due to ECDLP problem. The attacker $\mathcal{A}$ then generates a fresh private key t_{ai} $(< n)$ and computes the public key $A_i' = t_{ai}G$. $\mathcal{A}$ then sends $\langle SN_i, A_i' \rangle$ to node SN_j by replacing A_i by the distorted A_i' in the message. Node SN_j then generates secret key t_j $(< n)$, computes public value $A_j = t_jG$ over the elliptic curve $E_q(a, b)$, and sends $\langle SN_j, A_j \rangle$ to node SN_i.
$\mathcal{A}$ again intercepts the message $\langle SN_j, A_j \rangle$ and blocks that message. Since it is a computationally hard problem to find t_j from A_j for the attacker due to ECDLP problem, $\mathcal{A}$ then generates another private key t_{aj} $(< n)$, computes a public value $A_j' = t_{aj}G$ over the elliptic curve $E_q(a, b)$, and sends the modified message $\langle SN_j, A_j' \rangle$ to node SN_i by replacing A_j by the distorted A_j' in the intercepted message.
Now, node SN_i will generate the secret key $K_{i,a} = t_iA_j' = t_it_{aj}G$ shared with node SN_j. However, it is noted that in practice it is the secret key shared between node SN_i and attacker $\mathcal{A}$. In a similar way, node SN_j also generates the secret key $K_{j,a} = t_jA_i' = t_jt_{ai}G$ shared with SN_i, which is actually the key shared between node SN_j and attacker $\mathcal{A}$. Thus, the attacker $\mathcal{A}$ will be able to generate the above two keys $K_{i,a} = t_{aj}A_i$, which is the key shared with SN_i, and $K_{j,a} = t_{ai}A_j$, which is the key shared with SN_j. However, both nodes SN_i and SN_j still think that they are sharing only a single common secret key.
SN_i further computes $z_i = t_i + e_is_i \bmod q$, where $e_i = h(SN_i||K_{x_{ia}}||K_{y_{ia}})$, $K_{i,a} = (K_{x_{ia}}, K_{y_{ia}})$, $s_i = r_i + c_ix \bmod q$, $c_i = h(N_i||R_{x_i}||R_{y_i}||w_i)$, $R_i = r_iG = (R_{x_i}, R_{y_i})$, and sends the message $\langle z_i, R_i, w_i \rangle$ to node SN_j. $\mathcal{A}$ needs not to intercept the message $\langle z_i, R_i, w_i \rangle$. SN_j also sends $\langle z_j, R_j, w_j \rangle$ to SN_i, where $z_j = t_j + e_js_j \bmod q$, $e_j = h(N_j||K_{x_{ja}}||K_{y_{ja}})$, $K_{j,a} = (K_{x_{ja}}, K_{y_{ja}})$, $s_j = r_j + c_jx \bmod q$, $c_j = h(N_j||R_{x_j}||R_{y_j}||w_j)$, $R_j = r_jG = (R_{x_j}, R_{y_j})$. Again $\mathcal{A}$ needs not to intercept the message $\langle z_j, R_j, w_j \rangle$ and sends the same message to SN_i. When SN_i verifies the validity of conditions $w_j > T$ and $z_jG = A_j' + e_j(R_j + c_jQ)$, then the later verification does not hold. Thus, the signature verification fails and node SN_i will consider node SN_j as an illegitimate node. Similarly, node SN_j also verifies the conditions $w_i > T$ and $z_iG = A_i' + e_i(R_i + c_iQ)$. Again the later verification fails and as a result, the signature verification also fails and node SN_j will consider node SN_i as an illegal node.
The Huang's scheme can resist sybil attack, wormhole attack and node replication attack. However, this scheme does not have any ability to resist false reports injection attack and node capture attack.

F. Review of the Chatterjee et al.'s Scheme [9]

1) Description of the Protocol

Huang's scheme [33] is insecure against an active attack such as man-in-the-middle attack even though its authentication procedure and common key generation are simple and efficient. To eliminate such a serious attack, Chatterjee et al. [9] proposed an enhancement over Huang's access control scheme [33]. The authentication and key establishment phase of their scheme is different from that for Huang's scheme. The various phases of their scheme are discussed below.

a. Pre-deployment phase
Prior to deployment of sensor nodes in a deployment field, the CA first chooses a set of *network parameters* which includes (i) a finite field $GF(q)$, where q is a large odd prime of at least 160-bits; (ii) an elliptic curve $E_q(a, b)$; (iii) a base point G in $E_q(a, b)$ whose order is n; (iv) the CA's private key $x \in Z_n^*$, where $Z_n^* = \{1, 2, \ldots, n-1\}$; and (v) the CA's public key $Q = xG$. After that each deployed sensor node SN_i is assigned the bootstrapping time T_i and the deployment version DV_i. For each deployed sensor node SN_i, the base station generates a random number r_i and then computes $R_i = r_iP = (R_{x_i}, R_{y_i})$ and $s_i = r_i + c_ix \bmod q$ where $c_i = H(SN_i||R_{x_i}||R_{y_i}||T_i||DV_i)$. Finally, the base station preloads a set of *node parameters* for each SN_i prior to its deployment in the target field, which contains (i) its own unique node identifier SN_i; (ii) the elliptic curve $E_q(a, b)$; (iii) the base point P; (iv) the base station's public key Q; (v) a hash function $H(\cdot)$; (vi) DV_i; (vii) T_i; (vii) R_i; (viii) s_i; and (ix) the local variable called the latest version checked (lvc_i). The deployment version DV_i of a node SN_i is ini-

tialized as follows

$$DV_i = \begin{cases} 1, \text{ if } SN_i \text{ is deployed during initial deployment phase} \\ l, \text{ if } SN_i \text{ is deployed during } l\text{-th dynamic nodes addition phase.} \end{cases}$$

Initially, the value of lvc_i of node SN_i is assigned to the value of DV_i.

b. Authentication and key establishment phase

In this phase, after deployment each node first locates its neighbors in its communication range. Let SN_i and SN_j be two neighbors wants to establish a secret key between them after successful authentication. Node SN_i first generates a random nonce RN_i and a random secret number t_i ($< n$), which is its secret key. SN_i computes the public key $A_i = t_iP$ and the public value $z_i = t_i + s_i \bmod q$ over the elliptic curve, and sends the message $\langle SN_i, RN_i, T_i, DV_i, A_i, R_i, z_i\rangle$ to SN_j. When SN_j receives that message from SN_i, SN_j verifies the bootstrapping time T_i and deployment version DV_i of SN_i with its own T_j and DV_j. If $T_i = T_j$ and $DV_i = DV_j$, then SN_j considersSN_i as legitimate and also ensures that SN_j is deployed during the same deployment phase. In this case, both nodes SN_i and SN_j are considered as new nodes. Further, SN_j verifies its own lcv_j with DV_i received in the message. If $lvc_j = 1$, it means that SN_j is deployed during the initial deployment and if $lvc_j = l$, it means SN_j is deployed during the l-th dynamic node addition deployment phase. Now, if $T_j > T_i$ then SN_j verifies whether $DV_j > DV_i$ and $lvc_j \geq DV_i$. If both are valid, SN_i is considered as a legitimate node by SN_j. In such a case, SN_i is considered as old node and SN_j is new deployed node. Finally, if $T_j < T_i$, SN_j verifies whether $DV_j < DV_i$ and $lvc_j \leq DV_i$. If both conditions hold, SN_i is considered as a legitimate node by SN_j, and SN_j is considered as old node whereas SN_i is considered as a new deployed node.

For further verification SN_j computes $c_i = H(SN_i||R_{x_i}||R_{y_i}||T_i||DV_i)$. SN_j then checks the condition $z_iP = A_i + (R_i + c_iQ)$. If this condition holds, SN_j also generates a random secret number t_j ($< n$) as its own secret key, and computes the public key $A_j = t_jP$, $K_{ij} = t_jA_i = (K_{x_{ij}}, K_{y_{ij}})$ over the elliptic curve and $z_j = t_j + e_js_j \bmod q$, where $e_j = H(SN_j||K_{x_{ij}}||K_{y_{ij}})$. It then computes the symmetric secret key $SK_{ij} = H(SN_i||SN_j||T_i||T_j||DV_i||DV_j||RN_i||RN_j||z_i||z_j||K_{x_{ij}}||K_{y_{ij}})$ shared with SN_i. To ensure that SN_i will share the same secret key, SN_j uses the challenge-response protocol as follows. SN_j creates a puzzle message, say PM, computes the encrypted puzzle using its computed key SK_{ij} as $E_{SK_{ij}}(PM)$ and sends $\langle SN_j||RN_i||RN_j||T_j||DV_j||A_j||R_j||z_j||E_{SK_{ij}}(PM)\ ||H(SK_{ij}||PM||RN_i||RN_j||T_i\ ||T_j||DV_i||DV_j)\rangle$ to SN_i. SN_i first verifies the received random nonce RN_i in the message with its own previously generated random nonce for authentication with node SN_j. If it holds, SN_i verifies the bootstrapping time T_j and deployment version DV_j of SN_j with its own T_i and DV_i in a similar way as previously done by SN_j. SN_i then computes $K_{ji} = t_iA_j = (K_{x_{ji}}, K_{y_{ji}})$ and the same symmetric secret key SK_{ji} as $SK_{ji} = H(SN_i||SN_j||T_i||T_j||DV_i||DV_j||RN_i||RN_j||z_i||z_j||K_{x_{ji}}||K_{y_{ji}})$ shared with SN_j. Furthermore, SN_i computes $c_j = H(SN_j||R_{x_j}||R_{y_j}||T_j\ ||DV_j)$ and $e_j = H(SN_j||K_{x_{ji}}||K_{y_{ji}})$. Node SN_i then checks whether the condition $z_jP = A_j + e_j(R_j + c_jQ)$. If it holds, SN_j is accepted as a legitimate node by the node SN_i. In addition, to solve the puzzle, SN_i first decrypts the encrypted puzzle $E_{SK_{ij}}(PM)$ using its own computed secret key SK_{ji} and then retrieves the puzzle as $PM' = D_{SK_{ji}}(E_{SK_{ij}}(PM))$. It computes $H(SK_{ji}||PM'||RN_i||RN_j||T_i||T_j||DV_i||DV_j)$ using the retrieved puzzle PM', its own computed key SK_{ji}, its own previously generated random nonce RN_i and its timestamp T_i and deployment version DV_i. If this computed hash value matches with the incoming hash value received in the message, SN_i ensures that the node SN_j shares the same secret key with it. In this way, the secret key SK_{ij} is stored by both nodes in their memory for future secret communication between them. When each node SN_i authenticates and establishes secret keys with its all neighbors nodes, it updates its local version checked variable (lvc_i) as $lvc_i = lvc_i + 1$ in order to prevent attacks. The transmission of messages during the authentication and key establishment phase is summarized in Table 11 .

Table 11: Authentication and key establishment phase in Chatterjee et al.'s scheme [9].

Node SN_i	Node SN_j
$\langle SN_i, RN_i, T_i, DV_i, A_i, R_i, z_i\rangle \xrightarrow{\quad}$	
	$\xleftarrow{\langle SN_j\|\|RN_i\|\|RN_j\|\|T_j\|\|DV_j\|\|A_j\|\|R_j\|\|z_j \|\|E_{SK_{ij}}(PM)\|\|H(SK_{ij}\|\|PM\|\| RN_i\|\|RN_j\|\|T_i\|\|T_j\|\|DV_i\|\|DV_j)\rangle}$

c. Dynamic node addition phase

For new node deployment, the base station needs to preload the information as done in pre-deployment phase for other nodes. After deployment, the new node performs the above authentication and key establishment phase in order to authenticate and establish pairwise secret keys with its existing neighbor nodes in the network.

2) Cryptanalysis of the Protocol

In this scheme, after successful authentication the shared symmetric secret keys are established for each pair of neighbor nodes in the network. These secret keys are different for each pair of neighbor nodes and then used to secure communications. Thus, the external adversaries are prevented from injecting false reports into the sensor network.

If a sensor node is compromised, then an attacker has the ability to compromise secret keys of its neighbor nodes only. As in this scheme the established secret keys among sensor nodes are different throughout the network, the effect of a captured node does not essentially lead to compromise secure communications among non-compromised nodes in the network. As a result, this scheme is unconditionally secure against node capture attacks.

If an adversary intercepts the message transmitted from SN_i to SN_j, then due to difficulty of solving ECDLP it is computationally infeasible to change A_i and z_i in the intercepted message $\langle SN_i, RN_i, T_i, DV_i, A_i, R_i, z_i\rangle$. Similarly, the attacker does not have any ability to change A_j and z_j in the

reply message from SN_j to SN_i due to difficulty of solving ECDLP. As a result, the man-in-the-middle attack is prevented in this scheme.

This scheme is also secure against sybil attack, because this scheme uses the proper bootstrapping time and deployment version. New malicious deployed nodes are also prevented from falsifying the latest bootstrapping time and deployment version because for computing z_i for node SN_i, an attacker requires to compute $s_i = r_i + c_i x \mod q$, where $c_i = H(SN_i||R_{x_i}||R_{y_i}||T_i||DV_i)$, which is a difficult problem as x is the private key of the base station.

This scheme has further ability to withstand wormhole attack because for each sensor node SN_i only the base station can generate $s_i = r_i + c_i x \mod q$ and $z_i = t_i + s_i$, where $R_i = (R_{x_i}, R_{y_i})$ such that the signature verification $z_i P = A_i + (R_i + c_i Q)$ by its neighbor node SN_j holds and also the signature verification $z_j P = A_j + e_j(R_j + c_j Q)$ by SN_i holds, where $s_j = r_j + c_j x \mod q$, $z_j = t_j + e_j s_j$ and $e_j = H(SN_j||K_{x_{ij}}||K_{y_{ij}})$. As the private key x is only known to base station, so an attacker does not have any ability to forge the base station to deploy new malicious nodes in the sensor network.

VII. Simulation for Formal Security Verification of the Existing Access Control Schemes using AVISPA tool

In this section, we analyze the formal security of all the existing access control schemes using the AVISPA tool, called the Automated Validation of Internet Security Protocols and Applications.

A. AVISPA

AVISPA is a widely-accepted push-button tool for the automated validation of Internet security-sensitive protocols and applications [9], [47], [15]. AVISPA contains four back-ends: On-the-fly Model-Checker (OFMC), Constraint-Logic-based Attack Searcher (CL-AtSe), SAT-based Model-Checker (SATMC) and Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP). OFMC performs several symbolic techniques to explore the state space in a demand-driven way. CL-AtSe provides a translation from any security protocol specification written as transition relation in intermediate format into a set of constraints which are effectively used to find whether there are attacks on protocols. SATMC builds a propositional formula and then the formula is fed to a state-of-the-art SAT solver to verify whether there is an attack or not. Finally, TA4SP is a back-end which approximates the intruder knowledge by using regular tree languages.

Protocols analyzed by the AVISPA tool need to be specified in a language called HLPSL (High Level Protocols Specification Language). It is based on roles: basic roles for representing each participant role, and composition of roles for representing scenarios of basic roles. Each role is also independent from the other, getting some initial information by parameters, communicating with the other roles by channels. The output format (OF) of AVISPA is generated using one of the back-ends. When the analysis of a protocol has been successful (by finding an attack or not), the output describes precisely what is the result, and under what conditions it has been obtained. In the OF, the first printed section SUMMARY indicates whether the tested protocol is safe, unsafe, or whether the analysis is inconclusive, and the second section DETAILS either explains under what condition the tested protocol is declared safe, or what conditions have been used for finding an attack, or finally why the analysis was inconclusive. Other sections such as PROTOCOL, GOAL and BACKEND are the name of the protocol, the goal of the analysis and the name of the back-end used, respectively. After comments and statistics, the trace of an attack (if any) is also printed in an Alice-Bob format.

```
role alice (SNi, SNj, CA: agent,
   H, F: hash_func,
   Snd, Rcv: channel(dy))
played_by SNi
def=
local
State:nat,
Si,Sj,K,G,Ki,Kj:text,
Ti,Tj,Li,Lj,RNi,RNj,V:text
const  si_ki_private, sj_kj_private, k_private, alice_bob,
       bob_alice:protocol_id
init State:=0
transition
1.State=0 /\ Rcv(start) =|>
  State':= 1 /\ secret({Si,Ki},si_ki_private,SNi)
    /\ secret({Sj,Kj},sj_kj_private,SNj)
    /\ secret({K},k_private,CA)
    /\ Snd(SNi,Ti,Li,F(Si.G).F(Ki.G).F(Ki.
              H(SNi.Ti.Li.F(Si.G)).F(K.F(Ki.G))))
2.State = 1  /\ Rcv(SNj,Tj,Lj,F(Sj.G).F(Kj.G).F(Kj.
           H(SNj.Tj.Lj.F(Sj.G)).F(K.F(Kj.G))))=|>
  State':=3 /\ V':=F(F(Kj.H(SNj.Tj.Lj.F(Sj.G)).
  F(K.F(Kj.G))).G.H(SNj.Tj.Lj.F(Sj.G)).F(K.F(Kj.G)))

3. State=3    /\ not(V'=F(Kj.G))=|>
   State':=9

4.State = 3  /\ V'=F(Kj.G)=|>
  State':= 5 /\ RNi':=new()
          /\ Snd(SNj,SNi,{RNi'}_F(Si.F(Sj.G)))
    /\ witness(SNi,SNj,alice_bob,RNi')
    /\ request(SNi,SNj,alice_bob,RNi')
5.State = 5  /\ Rcv(SNi,SNj,RNi,{RNj'}_F(Sj.F(Si.G))) =|>
  State':= 7 /\ Snd(SNj,SNi,RNj)
end role
```

Figure. 4: Role of the initiator, sensor node SN_i for Zhou et al.'s scheme

B. Specifying the Protocols

We have implemented the existing access control schemes [72], [35], [32], [38], [33], [9] for the authentication and key establishment phase under AVISPA model checkers. In this paper, we only provide the detailed implementation of the Zhou et al.'s scheme [72] for convenience of better understanding. In this protocol model, there are two basic roles, alice and bob which represent the participants: sensor nodes SN_i and SN_j, respectively. In Figure 4, we have given the specification in HLPSL language for the role of the initiator, the node SN_i. SN_i first receives the start signal and changes its state from 0 to 1 and sends the message $\langle SN_i, T_i, L_i, P_i, C_i, c_i \rangle$ to SN_j with the $Snd(\,)$ operation. It then waits to receive the message $\langle SN_j, T_j, L_j, P_j, C_j, c_j \rangle$ from SN_j from the $Rcv(\,)$ action. Here, the type declaration $channel\,(dy)$ declares that the channel for the Dolev-Yao threat model (as described in Section IV) [22]. Hence the intruder has the ability to intercept, analyze, and/or modify messages transmitted over the insecure channel. When SN_i receives the message from SN_j it immediately changes its state to 3 and verifies if $V = C_j$ and if so, it sends the message $\langle SN_j, SN_i, EP_{K_{i,j}}[RN_i] \rangle$ to SN_j by generating a

```
role bob (SNi, SNj, CA: agent,
  H, F:hash_func,
  Snd, Rcv: channel(dy))
played_by SNj
def=
local
State:nat,
Si,Sj,K,G,Ki,Kj:text,
Ti,Tj,Li,Lj,RNi,RNj,V:text
const  si_ki_private, sj_kj_private, k_private, alice_bob,
       bob_alice:protocol_id
init State:=0
transition
1.State = 0 /\ Rcv(SNi,Ti,Li,F(Si.G).F(Ki,G).F(Ki.
          H(SNi.Ti.Li.F(Si.G)).F(K.F(Ki.G))))=|>
   State':= 2 /\ V':=F(F(Ki.H(SNi.Ti.Li.F(Si.G)).
  F(K.F(Ki.G))).G.H(SNi.Ti.Li.F(Si.G)).F(K.F(Ki.G)))

2.  State = 2   /\ not(V'=F(Ki.G))=|>
    State':= 8
3. State = 2   /\ V'=F(Ki.G)=|>
   State':= 4  /\ secret({Si,Ki},si_ki_private,SNi)
      /\ secret({Sj,Kj},sj_kj_private,SNj)
      /\ secret({K},k_private,CA)
      /\ Snd(SNj,Tj,Lj,F(Sj.G).F(Kj.G)
        .F(Kj.H(SNj.Tj.Lj.F(Sj.G)).F(K.F(Kj.G))))

4. State = 4  /\ Rcv(SNj,SNi,{RNi'}_F(Si.F(Sj.G)))=|>
   State':= 6 /\ RNj':= new()
            /\ Snd(SNi,SNj,RNi,{RNj'}_F(Sj.F(Si.G)))
            /\ witness(SNj,SNi,bob_alice,RNj')
     /\ request(SNj,SNi,bob_alice,RNj')
5.State = 6   /\ Rcv(SNj,SNi,RNj)=|>
  State':= 8
end role
```

Figure. 5: Role of the responder, sensor node SN_j for Zhou et al.'s scheme

random nonce RN_i at state 5. Finally, it waits to receive the message $\langle SN_i, SN_j, RN_i, EP_{K_{i,j}}[RN_j]\rangle$ from SN_j, sends the acknowledgment $\langle SN_j, SN_i, RN_j\rangle$ to SN_j at state 7 and terminates in this state.

In Figure 5, we have shown the specification in HLPSL language for the role of the responder, the node SN_j. After receiving the message $\langle SN_i, T_i, L_i, P_i, C_i, c_i\rangle$ from SN_i, SN_j sends the message $\langle SN_j, T_j, L_j, P_j, C_j, c_j\rangle$ to SN_i. Again, when SN_j will receive the message $\langle SN_j, SN_i, EP_{K_{i,j}}[RN_i]\rangle$ from SN_i, SN_j verifies the certificate and sends back the message $\langle SN_i, SN_j, RN_i, EP_{K_{i,j}}[RN_j]\rangle$ to SN_i.

The specifications in HLPSL language for the role of session and environment are specified in Figure 6. In the session segment, all the basic roles: alice and bob are instanced with concrete arguments. The top-level role (Environment) is always defined. This role contains global constants and a composition of one or more sessions, where the intruder may play some roles as legitimate user. The intruder (i) also participates in the execution of protocol as a concrete session.

The current version of HLPSL supports the standard authentication and secrecy goals. In the Zhou et al.'s scheme, the following three secrecy goals and two authentications are verified:

- secrecy_of si_ki_private: It represents that s_i and k_i are kept secret to SN_i.
- secrecy_of sj_kj_private: It represents that s_j and k_j are secret to SN_j.
- secrecy_of k_private: It represents that k is secret to the CA.
- authentication_on alice_bob: SN_i generates a random nonce RN_i where RN_i is only known to SN_i. If SN_j gets RN_i from the message from SN_i, SN_j authenticates SN_i.

```
role session(SNi, SNj, CA: agent,
        H, F: hash_func)
def=
local Sa, Ra, Sb, Rb:channel(dy)
const si_ki_private, sj_kj_private,
     k_private, alice_bob,
     bob_alice: protocol_id
composition
       alice(SNi,SNj,CA,H,F,Sa,Ra)
    /\ bob(SNi,SNj,CA,H,F,Sb,Rb)
end role
```

```
role environment()
def=
  const sni, snj, ca: agent,
        h, f: hash_func,
g: text,
        ti, tj, li, lj: text,
si_private, sj_private, k_private,
        alice_bob, bob_alice: protocol_id
intruder_knowledge = {sni,snj,ca,h,f,g,ti,
                tj,li,lj}
composition
   session(sni,snj,ca,h,f)
/\ session(snj,sni,ca,h,f)
end role

goal
secrecy_of si_ki_private
secrecy_of sj_kj_private
secrecy_of k_private
authentication_on alice_bob
authentication_on bob_alice
end goal
environment()
```

Figure. 6: Role of session and environment for Zhou et al.'s scheme

- authentication_on bob_alice: SN_j generates a random nonce RN_j, where RN_j is only known to SN_j. If SN_i receives RN_j from the message from SN_j, SN_i authenticates SN_j.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\
    Zhouetal_access_control.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.15s
  visitedNodes: 36 nodes
  depth: 6 plies
```

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  C:\progra~1\SPAN\testsuite\results\
    Zhouetal_access_control.if

GOAL
  As Specified

BACKEND
  CL−AtSe

STATISTICS

  Analysed   : 0 states
  Reachable  : 0 states
  Translation: 0.06 seconds
  Computation: 0.00 seconds
```

Figure. 7: The results of the analysis using OFMC and AtSe for the Zhou et al.scheme [72]

C. Analysis of Results

We have chosen the back-ends OFMC and Cl-AtSe for an execution test and a bounded number of sessions model checking [6]. For the replay attack checking, the back-ends check whether the legitimate agents can execute the specified protocol by performing a search of a passive intruder. After that the back-ends give the intruder the knowledge of some normal sessions between the legitimate agents [62], [47]. For the Dolev-Yao model check, the back-ends check whether there is any man-in-the-middle attack possible by the intruder.

We have simulated all the discussed existing schemes under both the back-ends OFMC and Cl-AtSe. The formal verification analysis of the Zhou et al.'s scheme shown in Figure 7 ensures that it is secure against replay and man-in-the-middle attacks.

We have simulated the Huang-Liu's scheme [35] under both the back-ends OFMC and Cl-AtSe. The formal verification analysis of the Huang-Liu's scheme shown in Figure 8 also

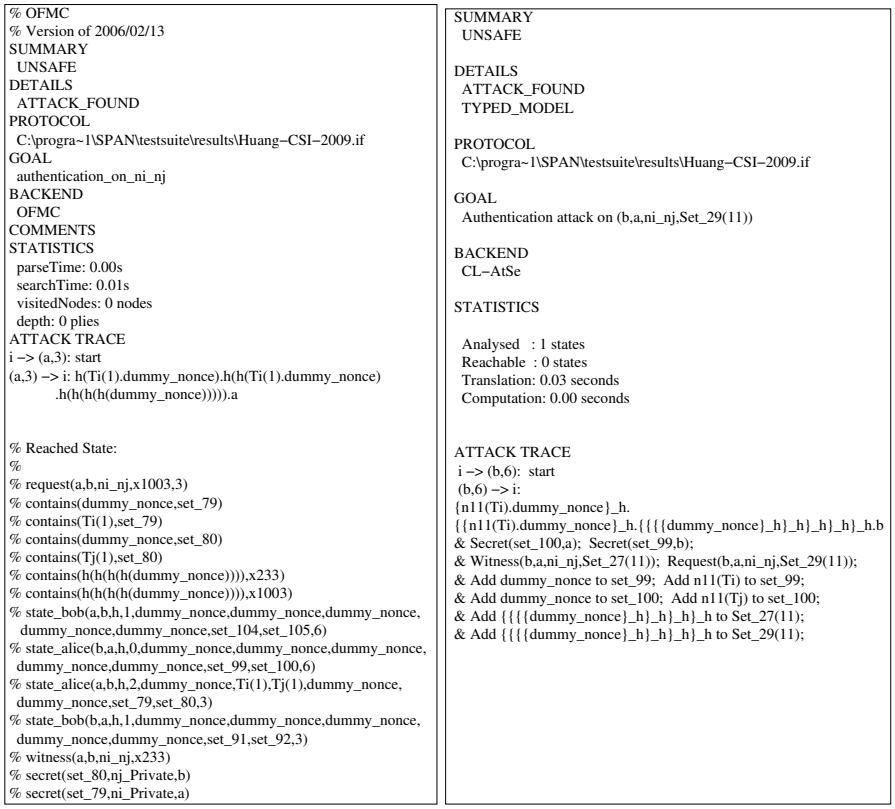

Figure. 9: The results of the analysis using OFMC and AtSe for the Huang's scheme [32]

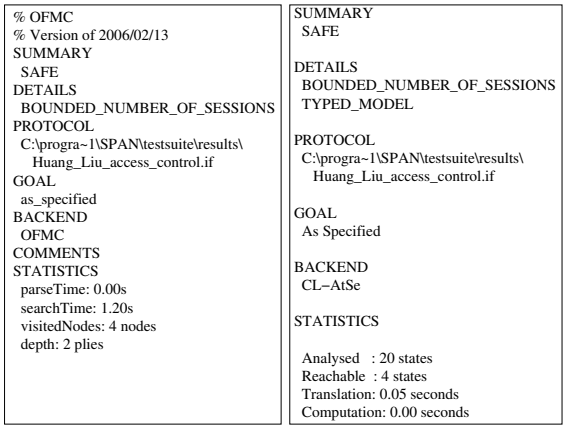

Figure. 8: The results of the analysis using OFMC and AtSe for the Huang-Liu's scheme [35]

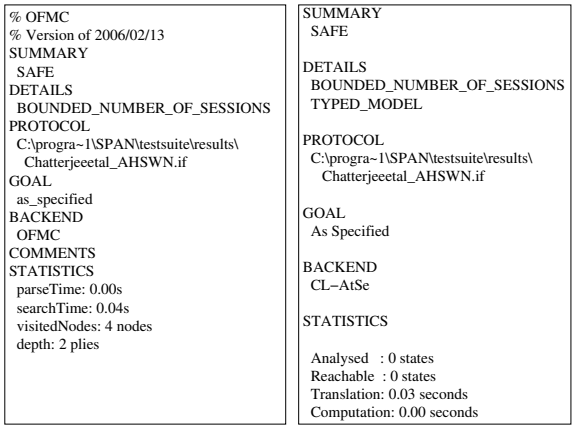

Figure. 10: The results of the analysis using OFMC and AtSe for the Chatterjee et al.'s scheme [9]

ensures that it is secure against replay and man-in-the-middle attacks.

We have analyzed the Huang's scheme [32] using the back-ends OFMC and Cl-AtSe. The results in Figure 9 show that this scheme is insecure against replay and man-in-the-middle attacks. This is clear from the attack traces produced by both OFMC and Cl-AtSe.

We have performed the formal security analysis of the recently proposed Chatterjee et al's scheme [9] under both the back-ends OFMC and Cl-AtSe. The analysis of the results available in Figure 10 ensures that this scheme is secure against both replay and man-in-the-middle attacks.

We have then simulated the Kim-Lee's scheme [38] under both the back-ends OFMC and Cl-AtSe. It is noted that the formal verification analysis of this scheme shown in Figure 11 ensures that it is unsafe against replay and man-in-the-middle attacks, which are evident from the attack traces available in this figure.

Finally, we have simulated the Huang's scheme [33] under both the back-ends OFMC and Cl-AtSe. It is clear that the formal verification analysis of this scheme shown in Figure 12 ensures that it is unsafe against replay and man-in-the-middle attacks, which are evident from the attack traces produced in this figure by the back-ends.

We have compared the results of the formal security analysis of all schemes for in Table 12. From this table it is clear that the Zhou et al.'s scheme, the Huang-Liu's scheme and the Chatterjee et al.'s scheme are safe, while the Huang's scheme

```
% OFMC
% Version of 2006/02/13
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\
    Enhanced−Huang−KimLee.if
GOAL
  authentication_on_ni_nj
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.01s
  visitedNodes: 0 nodes
  depth: 0 plies
ATTACK TRACE
i −> (a,3): start
(a,3) −> i: h(Ti(1).dummy_nonce).
h(h(Ti(1).dummy_nonce).h(h(h(h(dummy_nonce.Ai(1)))))).a

% Reached State:
%
% request(a,b,ni_nj,Ai(1),3)
% contains(dummy_nonce,set_79)
% contains(Ti(1),set_79)
% contains(dummy_nonce,set_80)
% contains(Tj(1),set_80)
% state_bob(a,b,h,1,dummy_nonce,dummy_nonce,dummy_nonce,
dummy_nonce,dummy_nonce,dummy_nonce,dummy_nonce,set_100,
set_101,6)
% state_alice(b,a,h,0,dummy_nonce,dummy_nonce,dummy_nonce,
dummy_nonce,dummy_nonce,dummy_nonce,dummy_nonce,set_97,
set_98,6)
% state_alice(a,b,h,2,dummy_nonce,Ti(1),Tj(1),dummy_nonce,
dummy_nonce,Ai(1),dummy_nonce,set_79,set_80,3)
% state_bob(b,a,h,1,dummy_nonce,dummy_nonce,dummy_nonce,
dummy_nonce,dummy_nonce,dummy_nonce,dummy_nonce,set_91,
set_92,3)
% witness(a,b,ni_nj,Ai(1))
% secret(set_80,nj_Private,b)
% secret(set_79,ni_Private,a)
```

```
SUMMARY
  UNSAFE

DETAILS
  ATTACK_FOUND
  TYPED_MODEL

PROTOCOL
  C:\progra~1\SPAN\testsuite\results\
    Enhanced−Huang−KimLee.if

GOAL
  Authentication attack on (b,a,ni_nj,n11(Ai))

BACKEND
  CL−AtSe

STATISTICS

  Analysed   : 1 states
  Reachable  : 0 states
  Translation: 0.03 seconds
  Computation: 0.00 seconds


ATTACK TRACE
 i −> (b,6):  start
 (b,6) −> i:  {n11(Ti).dummy_nonce}_h.
{{n11(Ti).dummy_nonce}_h.{{{{dummy_nonce.n11(Ai)}
  _h}_h}_h}_h}_h.b
& Secret(set_98,a);  Secret(set_97,b);
& Witness(b,a,ni_nj,n11(Ai));  Request(b,a,ni_nj,n11(Ai));
& Add dummy_nonce to set_97;  Add n11(Ti) to set_97;
& Add dummy_nonce to set_98;  Add n11(Tj) to set_98;
```

Figure. 11: The results of the analysis using OFMC and AtSe for the Kim-Lee's scheme [38]

```
% OFMC
% Version of 2006/02/13
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\
    Huang2011_access_control.if
GOAL
  secrecy_of_subs
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.00s
  visitedNodes: 1 nodes
  depth: 1 plies
ATTACK TRACE
i −> (sni,3): start
(sni,3) −> i: h(Ti(1).dummy_nonce).sni
i −> (sni,3): h(Ti(1).dummy_nonce).snj
(sni,3) −> i: f(Ti(2).f(h(sni.f(Ti(2)
.h(Ti(1).dummy_nonce)))
.f(dummy_nonce.h(sni.h(dummy_nonce.
dummy_nonce).dummy_nonce)
.dummy_nonce))).h(dummy_nonce.dummy_nonce)
.dummy_nonce
i −> (i,17): dummy_nonce
i −> (i,17): dummy_nonce


% Reached State:
%
% secret(dummy_nonce,subs,bs)
% secret(dummy_nonce,subs1,sni)
% secret(dummy_nonce,subs2,snj)
% witness(sni,snj,alice_bob_ti,Ti(1))
% state_bob(snj,sni,bs,h,f,0,dummy_nonce,
dummy_nonce,dummy_nonce,dummy_nonce,
dummy_nonce,dummy_nonce,dummy_nonce,dummy_nonce,6)
% state_alice(snj,sni,bs,h,f,0,dummy_nonce,
dummy_nonce,dummy_nonce, dummy_nonce,dummy_nonce,
dummy_nonce,dummy_nonce,dummy_nonce,6)
% state_alice(sni,snj,bs,h,f,2,Ti(2),Ti(1),
dummy_nonce,dummy_nonce,dummy_nonce,dummy_nonce,
dummy_nonce,dummy_nonce,3)
% state_bob(sni,snj,bs,h,f,0,dummy_nonce,dummy_nonce,
dummy_nonce,dummy_nonce,dummy_nonce,dummy_nonce,
dummy_nonce,dummy_nonce,3)
```

```
SUMMARY
  UNSAFE

DETAILS
  ATTACK_FOUND
  TYPED_MODEL

PROTOCOL
  C:\progra~1\SPAN\testsuite\results\
    Huang2011_access_control.if

GOAL
  Secrecy attack on (dummy_nonce)

BACKEND
  CL−AtSe

STATISTICS

  Analysed   : 7 states
  Reachable  : 5 states
  Translation: 0.03 seconds
  Computation: 0.00 seconds


ATTACK TRACE
 i −> (sni,3):  start
 (sni,3) −> i:
  {n1(Ti).dummy_nonce}_h.sni
  & Witness(sni,snj,alice_bob_ti,n1(Ti));

 i −> (sni,3):  {n1(Ti).dummy_nonce}_h.snj
 (sni,3) −> i:  {n2(Ti).{{sni.{n2(Ti).
          {n1(Ti).dummy_nonce}_h}_f}_h
.{dummy_nonce.{sni.{dummy_nonce.dummy_nonce}_h
.dummy_nonce}_h.dummy_nonce}_f}_f}_f.
{dummy_nonce.dummy_nonce}_h.dummy_nonce
& Secret(dummy_nonce,snj);
Secret(dummy_nonce,sni);
& Secret(dummy_nonce,bs);
```

Figure. 12: The results of the analysis using OFMC and AtSe for the Huang's scheme [33]

[32], the Huang's scheme [33] and Kim-Lee's scheme are unsafe.

Table 12: Summary of the results of analysis using OFMC and AtSe model checkers for existing schemes

Scheme	Results using OFMC and AtSe
Zhou et al. [72]	Safe
Huang-Liu [35]	Safe
Huang [32]	Unsafe
Kim-Lee [38]	Unsafe
Huang [33]	Unsafe
Chatterjee et al. [9]	Safe

VIII. Functionality Features and Performance Analysis of Different Access Control Schemes

In this section, we critically analyze the computational cost, communication cost and storage cost required for different existing access control schemes. We then thoroughly analyze the security analysis of different existing access control schemes. Finally, we make an overall comparison of functionality features and performance analysis of existing schemes.

A. Cost Analysis

1) Computational Cost

For analyzing the computational overhead for all schemes during the authentication and key establishment phase, we use the notations used in Table 13.

Table 13: Notations used for analysis of computational cost.

Symbol	Description
T_h	Time for performing a one-way hash function $H(\cdot)$, for example SHA-1
T_{enc}	Time for performing a symmetric-key encryption (AES encryption)
T_{dec}	Time for performing a symmetric-key decryption (AES decryption)
T_{ecm}	Time for performing a point-multiplication in elliptic curve $E_q(a,b)$
T_{eca}	Time for performing a point-addition in elliptic curve $E_q(a,b)$
T_{ecenc}	Time for performing an encryption using ECC
T_{ecdec}	Time for performing a decryption using ECC
T_{mul}	Time for executing a modular multiplication over finite field $GF(2^{163})$
T_{add}	Time for executing a modular addition in $GF(2^{163})$
T_i	Time for executing a modular inversion in $GF(2^{163})$

The point multiplication and modular inverse operations over an elliptic curve are computational expensive, whereas hashing computation is more efficient than those computations [33]. Moreover, elliptic curve encryption and decryption are computationally expensive as compared to those for symmetric key encryption and decryption (for example, AES encryption and decryption). In order to have a rough estimation of the computational complexity, we measure the computational cost of different access control schemes in terms of T_{mul} as in [68]. The rough estimation of different operations in terms of T_{mul} are given in Table 14.

Table 14: Time complexity of various operations in terms of T_{mul}

$T_{ecm} \approx 1200T_{mul}$	$T_{eca} \approx 5T_{mul}$	$T_i \approx 3T_{mul}$
T_{add} is negligible	$T_h \approx 0.36T_{mul}$	$T_{enc} \approx 0.15T_{mul}$
$T_{dec} \approx 0.15T_{mul}$	$T_{ecenc} \approx 2405T_{mul}$	$T_{ecdec} \approx 1205T_{mul}$

The quantitative analysis of [41] shows that the computation of a point multiplication requires approximately 1200 field multiplications; an elliptic curve point addition requires one field inversion and two field multiplications; the computation of a field inversion requires approximately three field multiplications; the computation of elliptic curve encryption and decryption require approximately 2405 and 1205 field multiplications respectively [56], [20]; and the cost of field addition is negligible. Further, it is noted that a 1024-bit modular multiplication takes 41 times longer than a field multiplication in finite field $GF(2^{163})$. The results of Wong et al. [66] show the speed for AES encryption and decryption, hash function using SHA-1 and 1024-bit modular multiplication and the results are $T_{enc} \approx 0.15T_{mul}$, $T_{dec} \approx 0.15T_{mul}$ and $T_h \approx 0.36T_{mul}$.

We compare the computational complexity using both formulated results and rough quantitative analysis in Table 15. From this table it is clear that Zhou et al.'s scheme is very expensive in terms of computational cost compared to other schemes during the authentication and key establishment phase. However, Huang-Liu's scheme requires minimum computational cost during the authentication and key establishment phase.

Table 15: Comparison of computational costs among different access control schemes.

Scheme	Formulated result	Rough estimation
[72]	$3T_{ecm} + T_i + T_h + 2T_{ecenc}/T_{ecdec}$	$7213T_{mul}$
[35]	$5T_h$	$2T_{mul}$
[32]	$2T_{ecm} + 4T_h$	$2401T_{mul}$
[38]	$2T_{ecm} + 9T_h$	$2409T_{mul}$
[33]	$5T_{ecm} + 4T_h$	$6001T_{mul}$
[9]	$4T_{ecm} + 4T_h$	$5401T_{mul}$

2) Communication Cost

For analysis of communication costs in terms of number of bits and number of packets required for different access control schemes, we use Table 16.

Table 16: Bit size of different parameters used in various access control schemes.

Type	Bit size
Node identifier (SN_i)	16
Bootstrapping time (T_i)	32
Length of bootstrapping phase (L_i)	16
Random number	32
Hash value (SHA-1)	160
Prime number (q)	160
ECC parameter, a	160
ECC parameter, b	160
ECC signature	320
Order of base point (n)	160
Expiration time (w_i)	32

Based on the number of bits used in different parameters, we have calculated the total number of bits required for all the messages during all the phases for each access control scheme. In order to calculate the number of packets required for transmission of a message during authentication and key establishment phase, and dynamic nodes addition phase for different access control schemes, we have used CC2420 transceiver [2]. CC2420 transceiver supports a packet of size 128 bytes, that is, 1024 bits. The results are shown in Table 17.

Table 17: Comparison of communication costs among different access control schemes.

Scheme	I_1	I_2
[72]	15232	20
[35]	1664	7
[32]	3904	10
[38]	4136	12
[33]	3392	8
[9]	4192	6

I_1 : Total number of bits transmission required for messages of all phases for schemes; I_2 : Total number of packets transmissions during authentication and key establishment phase, and dynamic nodes addition phase for schemes.

In wireless sensor networks, the transmission energy consumption rate approximately over three orders of magnitude greater than the energy consumption rates for computing [7]. From Table 17, it is also clear to note that the Zhou et al.'s scheme requires a lot of communication overhead compared with the Huang-Liu's scheme, the Huang's scheme, the Kim-Lee's scheme, the Huang's new scheme and the Chatterjee et al.'s scheme. Moreover, the Huang-Liu's scheme, the Chatterjee et al.'s scheme and the Huang's new scheme [33] outperform in term of communication overhead compared to the Zhou et al.'s scheme, the Huang's scheme and the Kim-Lee's scheme, because those schemes require a few number of packet transmissions only. However, the Huang-Liu's scheme, the Huang's scheme [32] and the Kim-Lee's scheme need the involvement of the base station during the authentication and key establishment phase too, whereas the Huang's new scheme [33], the Chatterjee et al.'s scheme and the Zhou et al.'s scheme [72] do not require to involve the base station during that phase.

3) Storage Cost

We have calculated the storage requirement for each sensor node required to store the necessary information for authentication and key establishment process prior to its deployment in the target field. We have again used Table 16 for calculating the storage requirements for different access control schemes. The results are then given in Table 18.

It is again noted that the Huang-Liu's scheme requires minimum amount of storage space for storing necessary information for a sensor node prior to its deployment as compared with that for other schemes. However, due to involvement of the base station after deployment of nodes in the network, the Huang-Liu's scheme [35], the Huang's scheme [32] and the Kim-Lee's scheme [38] require to store all the broadcasted information and renewable of hash chain (applicable for the Kim-Lee's scheme [38]) for all other nodes. As a result, the Huang-Liu's scheme [35], the Huang's scheme [32] and the Kim-Lee's scheme [38] are not scalable, whereas the Zhou et al.'s scheme [72], the Huang's new scheme [33] and the Chatterjee et al.'s scheme [9] are scalable as these schemes do not involve the base station after nodes deployment in the network for supporting a large-scale network.

Table 18: Comparison of storage costs among different access control schemes.

Scheme	Storage complexity required to store information prior to a node's deployment (in bits)
[72]	1824
[35]	496
[32]	1456
[38]	1616
[33]	1648
[9]	1664

Table 19: Comparison of various security attacks among different access control schemes.

Scheme	I_1	I_2	I_3	I_4	I_5	I_6	I_7
[72]	Yes	Yes	Yes	Yes	Yes	Yes	No
[35]	Yes	Yes	No	Yes	Yes	Yes	No
[32]	Yes	Yes	Yes	No	No	No	No
[38]	Yes	Yes	Yes	No	No	No	No
[33]	Yes	Yes	Yes	No	Yes	No	No
[9]	Yes	Yes	Yes	Yes	Yes	Yes	Yes

I_1 : Whether resists sybil attack or not; I_2 : Whether resists wormhole attack or not; I_3 : Whether resists node replication attack or not; I_4 : Whether resists man-in-the-middle attack/replay attack or not; I_5 : Whether resists new node masquerading attack or not; I_6 : Whether resists false reports injection attack or not; I_7 : Whether resilient against node capture attack or not.

B. Security Analysis

The security analysis of various access control schemes is shown in Table 19. It is noted that the Zhou et al.'s scheme is secure against several attacks excluding the resilience against node capture attack and the Chatterjee et al.'s scheme is highly secure against different attacks as compared to other existing schemes. All the schemes except the Chatterjee et al.'s scheme are not resilient against node capture attacks. However, the Huang's scheme [32], the Kim-Lee's scheme and the Huang's new scheme [33] are not resistant against active attacks such as man-in-the-middle attacks and replay attacks.

C. Overall Comparison

Finally, we have compared different access control schemes based on the four security requirements (SR1, SR2, SR3 and SR4) and seven functionality requirements (FR1, FR2, FR3, FR4, FR5, FR6 and FR7) defined in Section III. All these requirements for different existing access control schemes are summarized in Tables 20 and 21.

Table 20 shows that the Zhou et al.'s scheme protects SR1 (false reports injection attacks), SR2 (man-in-the-middle attacks) and SR4 (new node deployment attacks including malicious node deployment attack, sybil attack, node replication attack and wormhole attack), whereas their scheme is not secure against SR3 (resilience against node capture attack). The Huang-Liu's scheme protects SR1 and SR2, but their scheme is not secure against SR3 and SR4. The Huang's

scheme [32] and the Kim-Lee's scheme are not secure against all SR1, SR2, SR3 and SR4. The Huang's scheme [33] is secure against SR4, but it is insecure against SR1, SR2 and SR3. Finally, the Chatterjee et al.'s scheme protects against SR1, SR2, SR3 and SR4.

Table 20: Comparison of security requirements among different access control schemes.

Scheme	SR1	SR2	SR3	SR4
[72]	Yes	Yes	No	Yes
[35]	Yes	Yes	No	No
[32]	No	No	No	No
[38]	No	No	No	No
[33]	No	No	No	Yes
[9]	Yes	Yes	Yes	Yes

From Table 21, we see that all schemes support FR1 (dynamic node addition) after initial deployment of nodes such that existing deployed nodes do not require to change or update their stored information in order to authenticate and establish secret pairwise keys with new deployed nodes and FR2 (mutual authentication) for establishing secret keys between any two neighbor nodes in the network. The functionality requirement FR3 (network connectivity) is supported by all schemes, because all the schemes can establish pairwise secret keys with their neighbors after successful authentication. The Zhou et al.'s scheme requires high communication overhead (FR4) as compared to other schemes such as the Huang-Liu's scheme, the Huang's scheme [32], the Kim-Lee's scheme, the Huang's new scheme [33] and the Chatterjee et al.'s scheme [9]. The Zhou et al.'s scheme again requires high computational overhead (FR5) and storage overhead (FR6) as compared to other schemes. However, the Huang-Liu's scheme requires low communication overhead (FR4), computational overhead (FR5) and storage overhead (FR6) as compared to other schemes. Due to involvement of the base station during authentication and key establishment phase, dynamic node addition phase as well as renewable of hash chain phase, the Huang-Liu's scheme, the Huang's scheme [32] and the Kim-Lee's scheme are not scalable and hence they do not support a large number of sensor nodes in the network. On the other hand, the Zhou et al.'s scheme, the Huang's new scheme [33] and the Chatterjee et al.'s scheme are effective in supporting a large-scale network and as a result they are also scalable.

Table 21: Comparison of functional requirements among different access control schemes.

	FR1	FR2	FR3	FR4	FR5	FR6	FR7
[72]	Yes	Yes	High	High	High	High	High
[35]	Yes	Yes	High	Low	Low	Low	Low
[32]	Yes	Yes	High	Medium	Medium	Medium	Low
[38]	Yes	Yes	High	Medium	Medium	Medium	Low
[33]	Yes	Yes	High	Low	High	Medium	High
[9]	Yes	Yes	High	Medium	Medium	Medium	High

IX. Conclusion

In this paper, we have presented an overview of state of the art of access control protocols in wireless sensor networks available up-to date in the literature. We have defined the security and functionality requirements an ideal access control scheme should satisfy and achieve. We have described the existing access control schemes and based on the threat model, different attacks on schemes are presented in this paper which are done by other researchers and also done by us. We have analyzed thoroughly the communication cost, computational cost and storage cost required for each sensor node for access control in the network. We have then done the security and functionality comparison of schemes based on all defined security requirements and functionality requirements. We have shown that only the Zhou et al.'s scheme and the Chatterjee et al.'s scheme are secure against various attacks. The Zhou et al.'s scheme is not secure against the node capture attack, whereas all other schemes except the Chatterjee et al.'s scheme are insecure against most attacks. Though the Huang-Liu's scheme is very efficient in terms of communication, computation and storage requirements, it is insecure against various attacks. Due to involvement of the base station during authentication and key establishment phase, dynamic node addition phase and renewable of hash chain phase, the schemes (except the Zhou et al.'s scheme, the Huang's new scheme [33] and the Chatterjee et al.'s scheme) are not scalable in order to support large-scale sensor network. Moreover, the Zhou et al.'s scheme requires high communication, computation and storage requirements though it is more secure than most of the other existing schemes. Therefore, there is a need to look into the security and functionality goals in designing access control schemes in future research. Unfortunately, most of the schemes can not satisfy all the security requirements and achieve all the functionality requirements. Finally, we hope that our review work presented in this paper will provide a better understanding of the security challenges of access control, and pave the way for future research in this direction.

Acknowledgments

The authors would like to acknowledge the many helpful suggestions of the anonymous reviewers and the Editor-in-Chief of this Journal, which have improved the content and the presentation of this paper.

References

[1] Atmel Corporation. Available from: http://www.atmel.com. Accessed on September 2011.

[2] CC2420: 2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver. Available from: http://www.ti.com/ product/ cc2420. Accessed on September 2011.

[3] Secure Hash Standard. FIPS PUB 180-1, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, April 1995.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A Survey. *Computer Networks*, 38(4):393–422, 2002.

[5] J.P. Aumasson, L. Henzen, W. Meier, and M.N. Plasencia. Quark: A Lightweight Hash. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES 2010), LNCS*, volume 6225, pages 1–15, 2010.

[6] D. Basin, S. Modersheim, and L. Vigano. OFMC: A symbolic model checker for security protocols. *International Journal of Information Security.*, 4(3):181–208, 2005.

[7] D.W. Carman, P.S. Kruus, and B.J. Matt. Constraints and Approaches for Distributed Sensor Network Security. September 1, 2000. NAI Labs Technical Report # 010.

[8] H. Chan, A. Perrig, and D. Song. Random Key Predistribution Schemes for Sensor Networks. In *IEEE Symposium on Security and Privacy*, pages 197–213, Berkeley, California, 2003.

[9] S. Chatterjee, A. K. Das, and J. K. Sing. An Enhanced Access Control Scheme in Wireless Sensor Networks. *Ad Hoc & Sensor Wireless Networks*. , 2012 (in presss).

[10] T.-H. Chen and W.-K. Shih. A Robust Mutual Authentication Protocol for Wireless Sensor Networks. *ETRI Journal*, 32(5):704–712, Oct. 2010.

[11] A. K. Das. A Location-Adaptive Key Establishmen-t Scheme for Large-Scale Distributed Wireless Sensor Networks. *Journal of Computers*, 4(9):896–904, September 2009.

[12] A. K. Das. A Survey on Analytic Studies of Key Distribution Mechanisms in Wireless Sensor Networks. *Journal of Information Assurance and Security*, 5(5):526–553, 2010.

[13] A. K. Das. An Efficient Random Key Distribution Scheme for Large-Scale Distributed Sensor Networks. *Security and Communication Networks*, 4(2):162–180, 2011.

[14] A. K. Das. A random key establishment scheme for multi-phase deployment in large-scale distributed sensor networks. *International Journal of Information Security*, 11(3):189–211, 2012.

[15] A. K. Das. A secure and effective user authentication and privacy preserving protocol with smart cards for wireless communications. *Networking Science*, 2012. http://dx.doi.org/10.1007/s13119-012-0009-8.

[16] A. K. Das. Improving Identity-based Random Key Establishment Scheme for Large-scale Hierarchical Wireless Sensor Networks. *International Journal of Network Security*, 14(1):1–21, January 2012.

[17] 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*, 55(5):1646–1656, 2012.

[18] A.K. Das, A. Das, S. Mohapatra, and S. Vavilapalli. Key Forwarding: A Location-Adaptive Key-Establishment Scheme for Wireless Sensor Networks. In *International Workshop on Distributed Computing (IWDC 2005), LNCS 3741*, pages 404–409, 2005.

[19] M. L. Das. Two-Factor User Authentication in Wireless Sensor Networks. *IEEE Transactions on Wireless Communications*, 8(3):1086–1090, 2009.

[20] E. DeWin, A. Bosselaers, S. Vandenberghe, P. De Gersem, and J. Vandewalle. A fast software implementation for arithmetic operations in $GF(2^n)$. In *Proceedings of Advances in Cryptology - ASIACRYPT '96, Lecture Notes in Computer Science, Springer-verlag*, volume 1163, pages 65–76, 1996.

[21] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1976.

[22] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[23] J.P. Douceur. The Sybil attack. In *First International Workshop on Peer-to-Peer Systems (IPTPS'02)*, 2002.

[24] W. Du, J. Deng, Y. S. Han, and P. K. Varshney. A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. In *ACM Conference on Computer and Communications Security (CCS'03)*, pages 42–51, Washington DC, USA, October 27-31 2003.

[25] M. Eltoweissy, M. Moharram, and R. Mukkamala. Dynamic key management in sensor networks. *IEEE Communications Magazine*, 44(4):122–130, April 2006.

[26] L. Eschenauer and V. D. Gligor. A Key Management Scheme for Distributed Sensor Networks. In *9th ACM Conference on Computer and Communication Security*, pages 41–47, November 2002.

[27] R. Fan, L.-D. Ping abd J.-Q. Fu, and X.-Z. Pan. A Secure and Efficient User Authentication Protocol for Two-Tieres Wireless Sensor Networks. In *Second Pacific-Asia Conference on Circuits, Communications and System (PACCS 2010)*, pages 425–428, 2010.

[28] Y. Faye, I. Niang, and T. Noel. A Survey of Access Control Schemes in Wireless Sensor Networks. *World Academy of Science, Engineering and Technology*, 59(3):814–824, 2011.

[29] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In *Proceedings of 6th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'04)*, 2004.

[30] D. He, Y. Gao, S. Chan, C. Chen, and J. Bu. An Enhanced Two-Factor User Authentication Scheme in Wireless Sensor Networks. *Ad Hoc & Sensor Wireless Networks*, 10(4), 2010.

[31] Y. Hu, A. Perrig, and D.B. Johnson. Pachet leashes: a defense against wormhole attacks in wireless networks. In *IEEE INFOCOM'03*, 2003.

[32] H.-F. Huang. A novel access control protocol for secure sensor networks. *Computer Standards & Interfaces*, 31:272–276, 2009.

[33] H.-F. Huang. A New Design of Access Control in Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*, 2011, 2011. Article ID 412146, 7 pages doi:10.1155/2011/412146.

[34] H.-F. Huang, Y.-F. Chang, and C.-H. Liu. Enhancement of Two-Factor User Authentication in Wireless Sensor Networks. In *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 27–30, 2010.

[35] H.-F. Huang and K.-C. Liu. A New Dynamic Access Control in Wireless Sensor Networks. In *Proceedings of IEEE Asia-Pacific Services Computing Conference*, 2008.

[36] D. Johnson and A. Menezes. The Elliptic Curve Digital Signature Algorithm (ECDSA). Technical Report CORR 99-34, Dept. of C & O, University of Waterloo, Canada, August 23, 1999.

[37] M. K. Khan and K. Alghathbar. Cryptanalysis and Security Improvements of Two-Factor User Authentication in Wireless Sensor Networks. *Sensors*, 10:2450–2459, 2010.

[38] H.-S. Kim and S.-W. Lee. Enhanced novel access control protocol over wireless sensor networks. *IEEE Transactions on Consumer Electronics*, 55(2):492–498, 2009.

[39] L.-C. Ko. A Novel Dynamic User Authentication Scheme for Wireless Sensor Networks. In *IEEE International Symposium on Wireless Communication Systems 2008*, pages 608–612, 2008.

[40] N. Koblitz. Elliptic Curves Cryptosystems. *Mathematics of Computation*, 48:203–209, 1987.

[41] N. Koblitz, A. Menezes, and S. A. Vanstone. The state of elliptic curve cryptography. *Designs, Codes and Cryptography*, 19(2-3):173–193, 2000.

[42] B. Lai, S. Kim, and I. Verbauwhede. Scalable Session Key Construction Protocols for Wireless Sensor Networks. In *IEEE Workshop on Large Scale Real-Time and Embedded Systems*, 2002.

[43] X. H. Le, S. Lee, I. Butun, M. Khalid, R. Sankar, M. Kim, M. Han, Y.-K. Lee, and H. Lee. An Energy-Efficient Access Control Scheme for Wireless Sensor Networks based on Elliptic Curve Cryptography. *Journal of Communications and Networks*, 11(6), 2009.

[44] T.-H. Lee. Simple Dynamic User Authentication Protocols for Wireless Sensor Networks. In *The Second International Conference on Sensor Technologies and Applications, 2008*, pages 657–660, 2008.

[45] D. Liu and P. Ning. Establishing Pairwise Keys in Distributed Sensor Networks. In *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS)*, pages 52–61, Washington DC, Oct 27-31 2003.

[46] X. Liu, T. Huang, X. Wang, and X. Tang. A User Authentication Scheme based on Dynamic Password for Wireless Sensor Networks. In *2010 International Conference on Intelligent Computing and Integrated Systems (ICISS)*, pages 145–148, 2010.

[47] C. Lv, M. Ma, H. Li, J. Ma, and Y. Zhang. An novel three-party authenticated key exchange protocol using one-time key. *Journal of Network and Computer Applications*, 2012. http://dx.doi.org/10.1016/j.jnca.2012.04.006.

[48] D. J. Malan, M. Welsh, and M. D. Smith. A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. In *Proceedings of First IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON'04)*, Santa Clara, California, USA, 2004.

[49] J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis & defenses. In *3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, Berkeley, California, USA, 2004.

[50] R. W. D. Nickalls. A new approach to solving the cubic: Cardan's solution revealed. *The Mathematical Gazette*, 77(480):354–359, 1993.

[51] D.H. Nyang and M.-K. Lee. Improvement of Das's Two-Factor Authentication Protocol in Wireless Sensor Networks. In *Cryptology ePrint Archive*, 2009. Report 2009/631.

[52] B. Parno, A. Perrig, and V. Gligor. Distributed Detection of Node Replication Attacks in Sensor Networks. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy (S&P'05)*, 2005.

[53] R. D. Pietro, L.V. Mancini, Y. W. Law, S. Etalle, and P. Havinga. LKHW: A Directed Diffusion-Based Secure Multicast Scheme for Wireless Sensor Networks. In *Proceedings of 32nd International Conference on Parallel Processing Workshops (ICPPW '03)*, pages 397–406, 2003.

[54] R.L. Rivest, A. Shamir, and L.M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[55] C. P. Schnorr. Efficient identification and signatures for smart cards. In *Advances in Cryptology (Crypto '89), Lecture Notes in Computer Science*, volume 435, pages 339–351. Springer, 1990.

[56] R. Schroeppel, H. Orman, S. O'Malley, and O. Spatscheck. Fast key exchange with elliptic curve systems. In *Proceedings of Advances in Cryptology -*

CRYPTO '95, Lecture Notes in Computer Science, Springer-verlag, volume 963, pages 43 – 56, 1995.

[57] J. Shen, S. Moh, and I. Chung. COMMENT: "Enhanced Novel Access Control Protocol over Wireless Sensor Networks". *IEEE Transactions on Consumer Electronics*, 56(3):2019–2021, 2010.

[58] W. Stallings. *Cryptography and Network Security: Principles and Practices*. Prentice Hall, 3rd edition, 2003.

[59] H.-R. Tseng, R.-H. Jan, and W. Yangand. An Improved Dynamic User Authentication Scheme for Wireless Sensor Networks. In *IEEE GLOBECOM 2007 proceedings*, pages 986–990, 2007.

[60] B. Vaidya, D. Makrakis, and H. T. Mouftah. Improved Two-Factor User Authentication in Wireless Sensor Networks. In *Second International Workshop on Network Assurance and Security Services in Ubiquitous Environments*, pages 600–606, 2010.

[61] S. Vanstone. Responses to NIST's proposal. *Communications of the ACM*, 35:50–52, 1992.

[62] D. von Oheimb. The high-level protocol specification language hlpsl developed in the eu project avispa. In *In Proceedings of APPSEM 2005 Workshop*, 2005.

[63] H. Wang and Q. Li. Distributed User Access Control in Sensor Networks. *DCOSS 2006, Springer-Verlag, LNCS 4026*, 4026:305–320, 2006.

[64] Y. Wang, G. Attebuty, and B. Ramamurthy. A Survey of Security Issues in Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials*, 8(2):2–23, 2006.

[65] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus. TinyPK: securing sensor networks with public key technology. In *Proceedings of the 2nd ACM Workshop on Security of ad hoc and Sensor Networks, SASN 2004*, pages 59–64, Washington, DC, USA, October 2004.

[66] D.S. Wong, H.H. Fuentes, and A.H. Chan. The performance measurement of cryptographic primitives on palm devices. In *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC 2001)*, pages 92–101, 2001.

[67] K. Wong, Y. Zheng, J. Cao, and S. Wang. A dynamic user authentication scheme for wireless sensor networks. In *Proceedings of IEEE International Conf. Sensor Networks, Ubiquitous, Trustworthy Computing, IEEE Computer Society*, pages 244–251, 2006.

[68] S. Wu and K. Chen. An Efficient Key-Management Scheme for Hierarchical Access Control in E-Medicine System. *Journal of Medical Systems*, 36(4):2325–2337, 2012.

[69] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway. A survey of key management schemes in wireless sensor networks. *Computer Communications*, 30(11-12):2314–2341, 2007.

[70] J. Yuan, C. Jiang, and Z. Jiang. A Biometric-Based User Authentication for Wireless Sensor Networks. *Wuhan University Journal of Natural Sciences*, 15(3):272–276, 2010.

[71] P. Zeng, K.-K.R. Choo, and D.-Z. Sun. On the Security of an Enhanced Novel Access Control Protocol for Wireless Sensor Networks. *IEEE Transactions on Consumer Electronics*, 56(2):566–569, 2010.

[72] Y. Zhou, Y. Zhang, and Y. Fang. Access control in wireless sensor networks. *Ad Hoc Networks*, 5:3–13, 2007.

[73] S. Zhu, S. Setia, and S. Jajodia. LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks. In *Proceedings of 10th ACM Conf. Comp. and Commun. Security (CCS03)*, pages 62–72, 2003.

Author Biographies

Santanu Chatterjee is currently working as a Ph.D student in Computer Science and Engineering. He is also working in the Research Center Imarat (RCI), Defence Research and Development Organization, Hyderabad, India about two years. He received his M.Tech. degree in Computer Science and Engineering from the Jadavpur University, Kolkata, India. His research interests include cryptography and wireless sensor network security. He has published 4 papers in international journals.

Ashok Kumar Das is currently working as an Assistant Professor in the Center for Security, Theory and Algorithmic Research of the International Institute of Information Technology (IIIT), Hyderabad 500 032, India. He received his Ph.D. degree in Computer Science and Engineering, M.Tech. degree in Computer Science and Data Processing, and M.Sc. degree in Mathematics, all from the Indian Institute of Technology, Kharagpur, India, on April 2009, on January 2000 and July 1998, respectively. His current research interests include cryptography, wireless sensor network security, proxy signature, hierarchical access control and remote user authentication. He has published 39 papers in international journals and conferences in these areas.

Jamuna Kanta Sing is working as Associate Professor in the Department of Computer Science and Engineering, Jadavpur University, Kolkata, India. He received his Ph.D in Computer Science and Engineering from the Jadavpur University, Kolkata, India. He received M.Tech. in Computer Science and Engineering from the Indian Institute of Technology, Kharagpur, India and his B.Tech. in Computer Science and Engineering from the Jadavpur University, Kolkata, India. His current research interests include wireless sensor network security and medical image analysis. He has published many research articles in leading journals and conference proceedings.