A SECURE RANDOM KEY DISTRIBUTION SCHEME FOR INDUSTRIAL WIRELESS SENSOR SYSTEMS AGAINST NODE REPICATION ATTACKS

Safio, Snehal

Safio Sheik Abubakar Keilie, Department of Computer Engineering, RK University, Gujarat, India

And Prof senehal sathwara, Department of Computer Engineering, RK University, Gujarat, India

ABSTRACT

Since node replication attacks are conducted at the physical layer, they are difficult to defend against and have become increasingly common since the widespread adoption of wireless sensor networks in smart industrial systems. To address this issue, we propose a new method of defence against the attack by means of a secure random key distribution scheme (SRKD). To help identify and remove harmful nodes, we use a voting mechanism in conjunction with a locally tailored algorithm. To further protect against the replication attack, we alter the significance of the parameter s. In addition, when the number of network nodes reaches 200, the detection ratio of replicate nodes is greater than 90%, demonstrating the safety and efficacy of our scheme. When compared to other, more advanced schemes, SRKD shows superior storage and communication efficiency.

Keywords:- *wireless sensor system, Industrial IoT, node replication attack, random key distribution.*

1 INTRODUCTION

More and more vehicle networks, smart grids, and smart factories have adopted wireless sensor systems (WSS) [1]-[3], as have other applications of the IoT in industry, such as intelligent manufacturing [5], [6]. The WSS utilises a vast network of sensor nodes and is a type of distributed, multihop, self-configuring sensor system [7], [8]. Although the sensors' power, storage, and resources are constrained in deployed systems, they are still able to transmit data wirelessly [9]. WSSs are vulnerable to a variety of attacks [10], including hardware tampering, malicious message injection, and node replication, due to their use in high-value industrial applications. Attackers hope to reap "benefits" by compromising the entire system or a subset of its nodes. Industrial economies could suffer devastating losses, and the local community's safety and stability could be at risk if these attacks were to occur. Safeguarding WSS applications from hacking and other forms of data theft is crucial [11].

Motivation: When it comes to the WSS, a key management scheme is essentially used to establish resilience against node capture and information eavesdropping. Researchers have found that the random key distribution scheme is one of the most reliable secure key bootstrapping methods for WSS applications. Each node is provided with a random subset of keys from the key pool as part of a probability-based key management mechanism known as a random key distribution scheme. If two nodes in close proximity share a key, they will treat that key as the pairwise key and use it to create an encrypted connection between themselves. In the following paragraphs, we will discuss the most up-to-date methods for randomly distributing keys.

As a result, there is still a high risk that the data flow and honest nodes will be compromised, despite the fact that these cutting-edge schemes have so far proven resilient against node replication attack. In the case of insider attacks [12], for instance, a system attacker can impersonate captured nodes to create malicious replicas, which can then be deployed into the system and cause instability. Because the credentials of these malicious replicas are a replication of the compromised honest nodes, the replicas may be seen as "legitimate," which makes it difficult to design an effective defence mechanism against this attack.

What we've added: In this paper, we suggest a new method, dubbed SRKD. The following is an explanation of the novel aspects of our proposed SRKD. To handle duplicate nodes safely, we propose a novel approach. As a first step, we employ a voting mechanism to provide detection and replicas revocation in conjunction with the Efficient Distributed Detection (EDD) algorithm [12] for detecting node replication attacks. Then, to further aid in eradicating the replication attack, we alter the semantic significance of the parameter s. The maximum number of possible initial keys to establish a connection is referred to as "s" in this paper. As the likelihood of node replication increases if the number of shared starting keys between two nodes is equal to or greater than s, the link will not be established in this case. Next, to lessen the burden on the network brought on by the delivery of voting messages to the base station, we incorporate a message recovery mechanism into SRKD; this mechanism, full message recovery ID-based signature (MR-IBS), was proposed by Shim et al. [13].

When compared to other methods of random key distribution, ours is a significant improvement in both security and efficiency. When compared to competing replica detection methods, it fares admirably as well. The experimental outcomes prove our SRKD strategy can resist the node replication attack without compromising the integrity of the key bootstrapping process.

2 SYSTEM AND DEFINITION

When building an industrial wireless sensor network, it is common practise to make copies of already-existing devices for use in data backup, feature expansion, and other purposes. In Fig. 1 we see a schematic representation of a sensor system. Some trustworthy nodes are replicated within the system to increase the sensor cluster's processing power and the area over which it can detect changes.

Figure 2.1: System model.

Unfortunately, a malicious actor can use the data from captured good nodes to create new malicious replication nodes. The malicious copies can send bogus information into the system, which can have serious consequences.
Models of Attack After sensor deployment in SRKD, nodes are immediately vulnerable to compromise by the adversary. An adversary in the network has the ability to intercept all data. It can even trick the system into processing false information for its own benefit. While the adversary is unable to obtain the starting keys (ki), it is aware of their identifier (IDki) and the pairwise keys' (IDKshare) identifier. Information in the compromised nodes is immutable to the adversary. It has all the authentic credentials and initialization keys of the compromised nodes, though. In this way, it can use the data from compromised nodes to create new malicious nodes. There are clear distinctions between benign and malicious replicas, even though both types may copy data from the same trustworthy node. To a certain extent, the system permits and even encourages the use of benign replicas for ostensibly legitimate purposes. Malicious replicas, on the other hand, are un-trusted nodes that can compromise the entire network. Because of this, they cause irregularities in both data flow and user behaviour.

3 LITERATURE REVIEW

Es- chenauer and Gligor (EG) [14] came up with the idea of a random key distribution for the first time. Security in EG has been compromised. The q-Composite (QC) was proposed by Chan et al. [15] as a next-generation extension of EG. In QC, a pair of nodes will only be able to connect and build a communication channel if they have at least q starting keys in common and then generate a new pairwise key by hashing the concatenation of all of those keys. Even if a fraction of EG's nodes are compromised, QC remains more secure than its counterpart. However, the main limitation in QC's practical application is the larger amount of memory required. Gandino et al. [16] propose the q-s-Composite (QSC) as a refinement of QC. To restrict the possible combinations of keys, we use the upper bound parameter s, and to ease the burden of computation, we propose a method for quickly generating a pairwise key using wise XOR. Despite this, QSC is broken by the node replication attack.

The majority of distributed detection protocols for node replication attacks rely on the witness-finding technique to identify replicate nodes [17]. It is important to note that the underlying assumption of these methods is that each sensor node should send a signed location message to its neighbours [18]. For instance, in the randomised multicast (RM) scheme and the line-selected multicast (LSM) scheme proposed in [17], the witnesses are chosen at random. The similar but less expensive to communicate randomised, efficient, and distributed (RED) [19] scheme is based on the same idea. Also included in the category of witness-based schemes are the parallel multiple probabilistic cells (P- MPC) and the single deterministic cell (SDC) [18]. The plan in [20] is founded on the principle of the double ruling.

When it comes to mobile WSS, Yu et al. propose the first distributed detection algorithm to counter the node replication attack via a straightforward challenge-response strategy. The algorithm does not provide sufficient protection against collusive replicas, though.

Two localised algorithms, XED and EDD, based on the challenge-response and encounter-number strategy, are designed by Yu et al. [12] to overcome the restriction. Their goal is to significantly cut down on the amount of unnecessary communication.

4 OUR PROPOSED SCHEME

4.1 The Process of SRKD

We present a unique system known as SRKD in order to provide a mechanism for defending random key distribution against an attack known as the node replication assault. Algorithm 1 outlines both the process flow and the stages involved in the implementation of our plan. In the next portions of this section, the details of the algorithm will be expanded upon in greater detail.

4.2 Notations

To describe the proposed scheme, we first introduce some parameters that will be used later in the paper.

- n: the number of nodes in the network.
- e: the expected number of neighbour nodes for the node within communication range.
- n, : the number of neighbouring nodes.
- p: the number of starting keys in the key pool.
- q: the minimum number of shared keys that two neighbour nodes can establish a link.
- r: the number of starting keys of each ring that a node has.
- s: the maximum number of shared keys that two neighbour nodes can establish a link (Open interval).
- d: the expected number of links that a node can establish in key-setup phase.
- Tvote: the voting threshold that a node can be thought as a replication node.
- pro: the probability that two neighbouring nodes can establish a secure link during the key-setup phase.

Algorithm 1: Execution Process of SRKD.

```
1. SRKD: Execute the first deployment
2. Execute the keys distribution phase
3. Execute the keys establishment phase
4. Detect the node replication attack
5. while Node replication attacks exist in the system (the
   detection phase) do
6.   Execute the revocation of replicas (the revocation
     phase)
7. end while
8. Execute key updating periodically
9. Begin the next deployment
10. Execute the prevention (the prevention phase)
11. if The shared starting keys in nodes >= s then
12.   Revoke those nodes
13. else
14.   Execute the next keys distribution and establishment
15. end if
```

4.3 Random Keys Distribution in SRKD

1) The Keys Distribution Phase Each key has its own unique identifier, which is denoted by IDki.

Before the deployment of the network, the SRKD protocol chooses p keys at random from the key space to use in the establishment of the key pool. In addition, the starting keys are chosen at random by each node from among the r keys included in the ring. In addition, the base station will hand out a one-of-a-kind identification called an IDi to each node. This is done so that each node can save both the ring and their IDi.

2) Establishing the Keys Phase: During this phase, each node will periodically broadcast a handshake message to the network. The information that is stored in the ring's many nodes is depicted in Figure 4.1. The diagrams labelled Figure 4.2 to Fig. 4.3 each provide a concise explanation of the key establishment and the end state. When a node is the one to send the handshake message, we refer to it as the initiator. The identity (IDi) of the person who initiated the handshake is included in the handshake message, as are all of the IDkis of the keys (Ki) that are contained in the ring of the person who initiated the handshake. The handshake message is received by a node known as the receiver, which then confirms that the received IDki has those shared beginning keys.

In the event that there are fewer shared keys than the threshold value q, it will be impossible to create a connection between the two nodes, and the handshake will be necessary.

will be halted by the person receiving the message. When the number of shared keys is between q and s, the receiver will record the IDi of the initiator as well as the IDki of the shared keys. If the number of shared keys is between q and s, the receiver will not record this information. Due to the fact that the pairwise key must be generated each time before it can be utilised, SRKD does not keep the pairwise key. The pairwise key is determined by performing a bitwise XOR operation on the shared keys (Kshare = K1 K2 ...Ki...Kn. IDKshare is the name that is used to refer to it. The next step is for the receiver to relay an acknowledgement message to the sender. This communication includes the identity of the recipient as well as the IDki of the initial keys that have been shared. The acknowledgement of receipt message is authenticated using a MAC that is carried out by the paired key. The initiator is responsible for receiving and verifying the MAC before calculating the pairwise key. In the event that the results of the calculation are accurate, it will save the IDki of the shared keys received from the receiver with the identifiers (the position in the array). In the event that it is necessary to do so, the information can be put to use to calculate the pairwise key.

The key establishment phase is different from the QSC phase after the first set of nodes has been deployed. Within the context of this scenario, SRKD takes into consideration parameter s. If the number of shared keys discovered by the receiver is more than or equal to s, then the link between the two nodes will not be established. It is less likely that two nodes will have more than s similar keys due to the fact that all keys are taken randomly from a huge key pool and every node also picks the ring randomly from the key pool. As a result, we are of the opinion that the node replication attack takes place. It is possible for it to mitigate, to some extent, the damage that replicas may inflict on

the network and to cut down on the costs associated with the node revocation phase.

In addition, we implement a message recovery mechanism in SRKD in order to lessen the burden that is placed on the network's bandwidth. Please refer to [21] for further information about this topic.

4.4 The Detection Phase

During the phase that deals with the detection of EDD, we make an adjustment based on the detection technique that was proposed by Yu et al. [12].

1) Identification of the Node Replication Attack (EDD-Off-line Step): In the second algorithm, EDD does the calculations 1, 2, 12, 2, Y1 and Y2 in order to determine the value of that must be met before a node may be considered a replica. The count of times that a given node in the network, u, comes into contact with its adjacent nodes over a certain time interval, denoted by the notation L(u), is recorded. Regarding B(u), it is imperative that we highlight the fact that in our system, B(u) not only records the identity (IDi) of the replication node that is being examined by u, but it also contains the number of initial keys that are shared with I Di.

Algorithm 2 : EDD-Off-Line-Step

1. set $T = 1$, $B^{(\mu)} = \phi$, $\mu \in [1, n]$
2. set $\mathcal{L}^{(\mu)}[i] = 0$, $1 \leq i \leq n, \mu \in [1, n]$
3. **while** $Y_1 >= Y_2$ **do**
4. $T = T + 1$
5. calculate μ_1, μ_2, σ_1^2, σ_2^2
6. set $Y_1 = \mu_1 + 3\sigma_1$ and $Y_2 = \mu_2 - 3\sigma_2$
7. **end while**
8. set $\psi = \frac{Y_2 - Y_1}{2}$

2) The Detection of the Node Replication Attack (The EDD-On-Line Step): In order to carry out the EDD-On-line-Step, it is necessary for each node to periodically broadcast its IDi. Each node has its own local timer that it uses to keep track of the amount of time that has passed since the beginning of each time interval. The starter clock, known as timeist0. When the time interval T is exceeded, the result is t=0, which indicates the start of a new interval. The algorithm for EDD is denoted by the number 3, which is also its name.

As an additional line of defence against the node replication attack, the term B(u) is kept in the message that is transmitted to the base station.

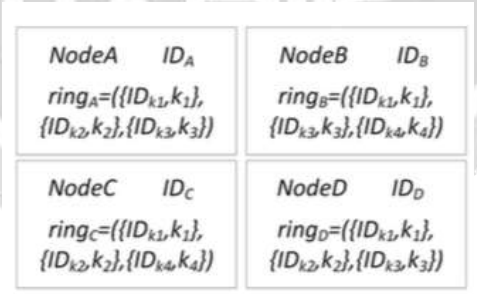


Figure 4.1: Information of ring in nodes.

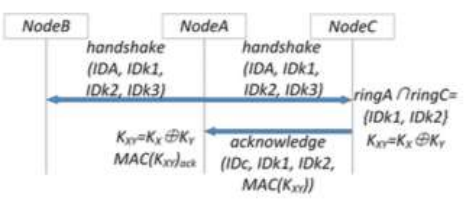


Figure 4.2: A-B-C handshake.

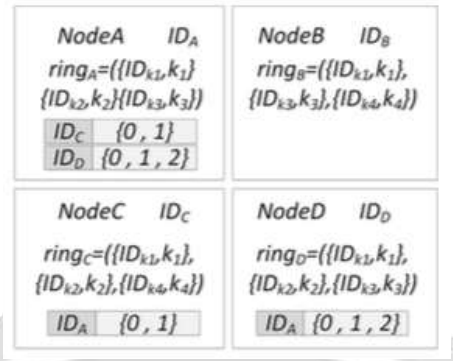


Figure 4.3: Pairwise key establishment.

Algorithm 3 : EDD-On-Line-Step

1. broadcast beacon b_u
2. **if** $t \neq t_0$ **then**
3. receive beacons b_{v_1},b_{v_d}
4. **while** $1 < \kappa < d$ **do**
5. $\mathcal{L}^{(\mu)}[v_\kappa] = \mathcal{L}^{(\mu)}[v_\kappa] + 1$
6. **end while**
7. **if** $\mathcal{L}^{(\mu)}[v_\kappa] > \psi$ **then**
8. set $\mathcal{B}^{(\mu)} = \mathcal{B}^{(\mu)} \cup \{v_\kappa\}$
9. **end if**
10. **else**
11. set $\mathcal{L}^{(\mu)}[s_\kappa] = 0$, $\kappa = 1 \ldots n$
12. **end if**

2) Voting: Each node will broadcast a check message to its adjacent nodes, and those nodes will send an acknowledge message back to the sending node as a return. The node that has been given an acknowledgement message will determine the number of nodes that are immediately adjacent to it, denoted by the notation n'. The value n′ as well as the total number of initial keys are placed into the ring r by the node. After that, it inserts the value B(u) into the message m and sends the message to the base station so that it can be voted on.

The value of d is kept up to date by the base station, which computes the d(di) of each node (IDi) using the formula pro= d, n= r, and n= (rn′).

Let's define the voting threshold using the notation Tvote = minimum, minimum (di) minimum. The base station will then count all of the bu that have been received and will update the total number of votes for each node. If the count for IDi is higher than Tvote, then the base station will use IDi as a replication node in its network. After then, do out the work required to revoke the sentence.

4.5 The Revocation Phase

During this stage of the process, the base station will broadcast the IDi of the replication nodes as well as the IDki of the keys of the ring contained within the replicas. The replication nodes' nearby nodes will severe their connection to the replicas and prohibit the use of shared keys with replication nodes. The base station is going to make it such that the IDki of the ring cannot be used in replication nodes. In addition to that, using these IDki in the pool is strictly prohibited.

Additionally, the system recovery is of the utmost significance. As the number of replicated nodes increases, the base station will prohibit the use of too many keys, which will result in a decrease in the number of keys that are accessible in the key pool. Because of this circumstance, the connectivity of the sensor system will be disrupted. Eliminating such malicious replication nodes is necessary if we are to stop the decrease in the quantity of keys from disrupting the connectivity of node systems. Due to the fact that the nodes are physically placed, it will be necessary for us to remove malicious nodes manually. Once those malicious replicas have been eliminated, the operator will notify the base station, which will then enable the keys that had previously been disabled.

The value of s is then brought up to date by SRKD. SRKD is responsible for calculating the maximum value by counting all shared keys across all replication nodes using IDi in each and every session. The maximum value will serve as the new value for s and will be implemented in the subsequent deployment in order to thwart any additional node replication attacks.

4.6 The Prevention Phase

Establishing a prevention mechanism against the node replication attack is necessary if we are to reduce the amount of damage that will be caused to the sensor system as a result of the node replication attack. In order to accomplish this, we make use of the fundamental aspects of communication between nodes. Our SRKD will not establish the link between the two nodes during the subsequent deployment if the number of shared keys is larger than or equal to the new s. Additionally, the base station will revoke the malicious replicas if the number of shared keys is equal to or equal to the new s. It is less rare for two nodes to share more than s keys due to the fact that each node chooses the ring to be used from the key pool in a random fashion. As a result, we are of the opinion that the node replication attack takes place. This procedure has the potential to mitigate, to some extent, the negative effects that replicas may have on the network.

5 PERFORMANCE ANALYSIS

5.1 Resilience

In this section, we conduct an analysis of the robustness of the capability to combat the presence of secret information that has been compromised.

The following formula can be used to determine the chance that an adversary can eavesdrop on a link between nodes without compromising the link's integrity.

QC link [15]:

$$[\sum_{i=q}^{r}\binom{r}{i}\binom{p-r}{r-i}\binom{p}{r}^{-x} \sum_{j=0}^{i}\binom{i}{j}(-1)^{j}\binom{p-j}{r}^{x}] /[\binom{p}{r}-\sum_{j=0}^{q-1}\binom{r}{j}\binom{p-r}{r-j}] \quad (1)$$

QSC link [16]:

$$[\sum_{i=q}^{r}\binom{r}{i}\binom{p-r}{r-i}\binom{p}{r}^{-x} \sum_{j=0}^{\mathrm{Min}(i,s)}\binom{Min(i,s)}{j}(-1)^{j}\binom{p-j}{r}^{x}] /[\sum_{k=q}^{r}\binom{r}{k}\binom{p-r}{r-k}] \quad (2)$$

Our SRKD link:

$$[\sum_{i=q}^{s}\binom{r}{i}\binom{p-r}{r-i}\binom{p}{r}^{-x} \sum_{j=0}^{\mathrm{Min}(i,s)}\binom{Min\,(i,s)}{j}(-1)^{j}\binom{p-j}{r}^{x}] \quad (3)$$
$$/[\sum_{k=q}^{r}\binom{r}{k}\binom{p-r}{r-k}]$$

The likelihood of a forged node successfully passing the authenticity validation in QC, QSC, and SRKD is defined by the following formula.

$$1-\sum_{j=1}^{q} j\binom{r}{q=j}(-1)^{j}\binom{p-r+q-j}{r}^{x}/\binom{p}{r}^{x} \quad (4)$$

The resistance against the eavesdropping of confidential information that is supplied by our proposed SRKD can be thought of as a general instance of that which is offered by QC, in the same way that the QSC scheme works.

The following are the components that make up the formula:

(1) There is a complete compromise of the Min(i, s) beginning keys that are used by the link in the rings that contain x nodes. The following equation can be used to calculate this situation:

The formula of

$$\binom{p}{r}^{-x}\sum_{j=0}^{Min(i,s)}\binom{Min\,(i,s)}{j}(-1)^{j}\binom{p-j}{r}^{x}$$

equals 1 when j = 0;

2) It should deduct the likelihood that at least one key for the connection is not included in the rings of x compromised nodes, and this probability should be tied to the iterations of the summation. This probability should be related to the iterations of the summation. In addition, the redundant items at the subsequent rounds of the summing are adjusted by alternatively deducting from or adding to the combination of (-1)j. This is done in order to eliminate the redundant items.

(2) Increase the likelihood that communication between two nodes that have shared I keys can be intercepted. Because q is more than I and s in our suggested system, the pairwise key is made up of starting keys that begin with the letters I and s. The likelihood is dependent on either of the following two scenarios:

Both SRKD and QSC produce the identical results with regard to the probability that a forged node will pass the authenticity confirmation shown by the formula. An adversary may pass the authenticity confirmation in either of the two schemes if they have a minimum of q beginning keys in common with a node and have established a link with that node. The calculation of the formula for SRKD is one minus the probability that in the node that would perform the authenticity confirmation, there are fewer than q starting keys in the ring that are shared with the x compromised nodes. This probability is calculated by subtracting one from the probability that there are less than q starting keys in the ring. The following is how it is computed:

Determine the number of instances where q fewer than j keys are shared in each and every iteration of the process.

Yu et al. [12] has proved the security and practicability of EDD by demonstrating its resilience to an attack that involves node replication. In addition, we improve the resistance technique by incorporating a voting mechanism to create a second confirmation, which has the potential to boost the accuracy of examination. This is done while

taking into account network latency, packet loss, and other environmental factors. The value min,min(di), which is the threshold in the voting mechanism, and min (di), which refers to the minimum expected number of links that a node can establish in the key-setup phase, are both referred to as "min." It's possible that some of the nodes in the network will experience a circumstance known as di, which makes it difficult for copies to be found using EDD. As a result, the vote threshold that we utilise to reply to the circumstance of di is min, min (di). In this scenario, we are in a position to form an opinion regarding the presence of malicious copies if nodes obtain more votes than di.

After that, we conduct an investigation into the safety of the SRKD key distribution phase by utilising the "Security Protocol Animator for Automated Validation of Internet Security Protocols and Applications" (SPAN+AVISPA)(SA) [22] and a role-oriented language known as high level protocol specification language (HLPSL).

Algorithm 4 outlines the steps that must be taken in order to validate SRKD using SA. Please refer to [21] for further information about this topic.

Figure 5.1 depicts the outcome of the experiment. The suggested SRKD is simulated by the OFMC backend, which is an excellent verification paradigm that can evaluate things like the Dolev-Yao model and replay attack, among other things. The indicator that can be seen in the result of the study reveals that our plan is "SAFE."

5.2 Detection and Revocation Evaluation of Malicious Replication Nodes

OMNet++ is the tool that we use to model our plan. There are 50, 100, 150, and 200 nodes in our simulated systems, according to the results of our simulation. After that, we inject twenty clones into each system in order to determine how effective our technique of defence is. The findings provide evidence that our proposed strategy is capable of withstanding attacks that involve node replication to a significant degree.

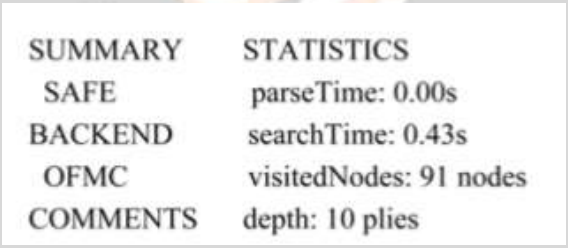


Figure 5.1: Experimental result under OFMC backend.

Our system is able to defend itself against attacks that include node replication in three different ways: detection, revocation, and prevention of replication nodes. In this part, we conduct an analysis of the detection and revocation processes in relation to malicious replication nodes. Both Figure 5.1 and Table 5.1 illustrate the number of replicas found in networks with varying numbers of nodes.

Algorithm 4: Verifying the SRKD in SPAN+AVISPA.

1. Simplify the communication in the system to the communication between two nodes
2. PART 1: Role Definition: NodeA/B
3. Declaration: agents in the system, parameters stored in NodeA/B, encryption key, hash function, transmission channel
4. Definition: local network environment, initial state of network
5. Transition: list of network state transitions
6. PART 2: Role Definition: Session
7. Declaration: agents in the system, parameters stored in agents, encryption key, hash function
8. Definition: local network environment, transmission channel
9. Composition: the agents composition of the session
10. PART 3: Role Definition: Environment
11. Definition: local network environment, transmission channel, parameters stored in agents in the system environment, encryption key, hash function, security parameters
12. Initialization: parameters stored in agents in the system environment, intruder knowledge
13. Composition: the sessions composition of the environment
14. PART 4: Role Definition: Goal
15. Safety Expectation: security parameters
16. PART 5: Run Environment

It is clear that the accuracy of detection improves along with the proportion of normal nodes that are present in the network. We also discover that if there are only a few nodes in the network, the threshold of EDD may be higher than the minimum (di). Due to the fact that the node at the edge of the network that has the value d = min (di) is not required to construct at least min(di) links, a replica is unable to receive more votes than the value min (di). The probability of occurrence of nodes (d = min (di)) decreases as the number of nodes in the network increases (as the number of nodes in the network increases). As a direct consequence of this, the probability that replicas will be set up in close proximity to those nodes (d = min (di)) will drop.

Table 5.1: The Detection Result of SRKD

Num of nodes	50	100	150	200
Num of replicas	20	20	20	20
Num of replicas detected (Expt.1)	14	16	16	19
Num of replicas detected (Expt.2)	15	19	18	18
Num of replicas detected (Expt.3)	13	15	16	20
Num of replicas detected (Expt.4)	16	17	19	19
Num of replicas detected (Expt.5)	14	16	19	20
Normal node ratio	71.43%	83.33%	88.24%	90.91%
Avg detection ratio	72%	83%	86%	96%

Table 5.2 presents the detection rates achieved by various cutting-edge detection methods that are currently in use (N denotes the number of nodes). When there are fewer than one hundred nodes in a network, it is clear that our SRKD has a detection rate that is about equivalent to that of a medium-level system. On the other hand, SRKD delivers a satisfactory performance when the number of nodes exceeds 200. Even though TDD and SDD provide the impression of having a perfect solution, this is only demonstrated in a few specific instances. Both TDD and SDD are affected by the efficiency limitation in some way. The most important aspect of our SRKD scheme is that it is based on random key distribution, and it is simple to combine with other schemes that utilise random key distribution.

Regarding the revocation, we make the assumption that the replication nodes, once found, may be successfully removed from the network without causing any problems.

Table 5.2: The Comparison of Detection Rates of Different Schemes

Scheme	N< 100	N> 200	Scheme	N< 100	N> 200
RM [17]	63%	95%	P-MPC [18]	86%	89%
LSM [17]	55%	90%	TDD [20]	-	100% particular case
RED [19]	87%	90%	SDD [20]	-	100% particular case
LINE [18]	62%	74%	EDD [12]	70%	95%
SDC [18]	70%	95%	SRKD	72%	96%

5.3 Prevention Evaluation of Malicious Replication Nodes

In the prevention section, the reason that we use the max (s) is to guarantee the normal deployment of genuine nodes by suitably limiting the accuracy of detection. This is done so that we can guarantee the normal deployment of genuine nodes. Figure 6 and Figure 7 show, respectively, the number of clones that were stopped from being created and the number of genuine nodes that were mistakenly eliminated. The threshold of max (s) can encompass a greater region and can better represent the generality of all the nodes in the network if there are more nodes in the network. As a result, it has the potential to cut down on the number of false positives. However, as max (s) grows higher, certain replicas that have not shared more than max (s) starting keys cannot be stopped. This number of replicas is dependent on the max (s) value. In the future, this particular aspect has to be enhanced.

5.4 Other Evaluation

1) Storage Efficiency Evaluation In this section, we evaluate the efficiency of storing information using SRKD and compare it to the efficiency of storing information using EG, QC, and QSC. Additionally, we evaluate the efficiency of storing information using SRKD. The evaluation will take into account the following criteria and weights:

Table 5.3 presents the costs associated with using memory. The primary component of the needed amount of storage space in the prestorage of EG is designated for the safekeeping of r-keys. Memory needs to be accessed in order to save not only the starting keys but also the paired keys for the pre-storage phase of QC. A hash function is also stored within QC. However, it does not save any identifier of the paired key since, in the future, the neighbour nodes will utilise the shared keys to form a new link. This is why it does not retain any identifier of the pairwise key. The storage of lk lkID is what differentiates EG and QC with regard to the working storage portion of the system. In addition, the memory requirements for lk are higher than those for lkID. It keeps the initial keys, those IDs, the identification of the node, as well as the key for updating all of this information in the prestorage of QSC. The number of identifiers of those keys used to establish the paired key is designated as maximum (maxrg) in the working storage of QSC, which can give a pessimistic examination for QSC. This is because the maximum number of identifiers is used to establish the pairwise key.

However, the working storage in SRKD is not the same as it is in QSC. The pre-storage in both systems is the same. In SRKD, the memory demand for L(u) is denoted by vlnum, while the memory requirement for B is denoted by maxrp (lID + lnum) (u). In the same line of thought as maxrg, SRKD considers maxrp to be the number of neighbour nodes that have been taken into account as replicas. Both the number of identifiers and the number of copies can be reduced below the maximum allowed (maxrg and maxrp, respectively).

Table 5.3: Comparison in Memory Cost

Protocol	Pre-storage	Working storage
EG [14]	$r*(l_k+l_{kID})$	$r*(l_k+l_{kID})+$ $v*(l_{ID}+l_{kID})$
QC [15]	$r*(l_k+l_{kID})$	$r*(l_k+l_{kID})+$ $v*(l_{ID}+l_k)$
QSC [16]	$r*(l_k+l_{kID})+$ l_k+l_{ID}	$r*(l_k+l_{kID})+v*$ $(l_{ID}+maxrg*l_{kID})+$ l_k+l_{ID}
SRKD	$r*(l_k+l_{kID})+$ l_k+l_{ID}	$r*(l_k+l_{kID})+v*$ $(l_{ID}+maxrg*l_{kID})$ $+l_k+l_{ID}+l_T+l_\psi+$ $v*l_{num}+$ $maxrp(l_{ID}+l_{num})$

For the convenience of a comparison, we consider the following case used in QSC: r = 10, v = 10, maxrg = 5, maxrp=5,lk =16bytes,lID =2bytes,lkID =1byte, lT = 1 byte, lψ = 1 byte and lnum = 1 byte. The comparison is shown in Fig. 9.

EG is the first random key distribution method, and while it has a low storage overhead, the level of security it offers is insufficient. QC makes up for the lack of safety, but at the expense of consuming an excessive amount of storage space. On the basis of QC, QSC makes significant improvements to the storage efficiency. From Figure 9, we can deduce that the storage requirements for SRKD are significantly lower than those for QC. Even while it has somewhat greater storage requirements than QSC, the increase in storage requirements is still well below the permitted bounds for security advances (against a replication attack).

2) An Analysis of the Effectiveness of Communication: In this section, we will analyse the communication burden that SRKD places on us. In EDD, the length of the beacons that are required to be exchanged is denoted by lb, and the length of the signature is denoted by l, hence the equation for the length of the signature is l = lb (R, y, z).

In order to establish a link using an EG, QC, QSC, or SRKD scheme, it is necessary to perform two transmissions of a single hop each. The identifier of the node, known as lID, as well as the identifiers of the keys in the ring, known as rlkID, are both stored in the handshake message used in SRKD. The identifier of the sender of the handshake message lID, the message authentication code for the handshake message lk, and the identifiers of the selected keys maxrg and lkID that are used to establish the link are all stored in the acknowledge message. A pessimistic analysis is performed with the assistance of the maxrg.

Table 5.4 outlines, for each of several distinct approaches, how effective the communication is. Communication in SRKD can take place in one of two ways: either "node to node" or "node to base station." "node to node" communication is the more common of the two. We assume that r is equal to 10, maxrg is equal to 5, lk is equal to 16 bytes, lID is equal to 2 bytes, lkID is equal to 1 byte, l is equal to 41 bytes, and lm is equal to 17 bytes. In our method for recovering messages, the point on the elliptic curve that represents the message is specified over the finite field Fq, with q equal to 20.5 bytes (For ECC, a general safety requirement is approximately 200-bit). The point on the elliptic curve is also represented by the equation Q = when affine coordinates are used (x,y). The signature for SRKD is written as = (R, y, z), and it consists of two values of the coordinate in Zq as well as a point on an elliptic curve. During the Extract phase, R is, on the other hand, fed into the sensor node beforehand. Because of this, the length of the expression = (R, y, z) l may be determined as follows: || = |y|+|z| = 20.5 bytes+20.5 bytes = 41 bytes The message that is transmitted from the node to the base station includes the value B(u) as well as the identify of the node. In light of this, the formula for lm is as follows: lm = maxrp (lID + lnum) + lID

Table 5.4: Communication Comparison

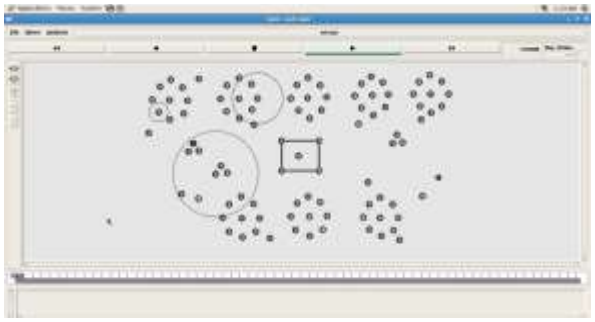
Scheme	Transmission size
EG [14]	$(r+1)*l_{kID}+2*l_{ID}+l_k$
QC [15]	$2r*l_{kID}+2*l_{ID}+l_k$
QSC [16]	$(r+maxrg)*l_{kID}+2*l_{ID}+l_k$
SRKD	$(r+maxrg)*l_{kID}+2*l_{ID}+l_k+l_b$

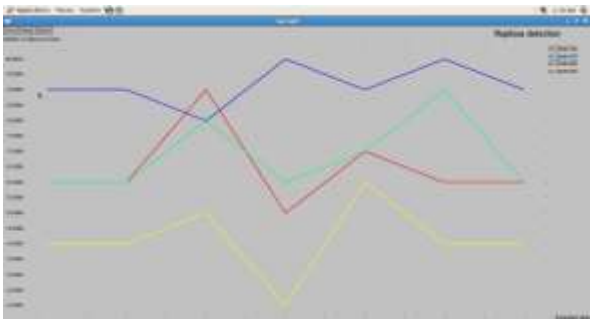
Table 5.5: Communication Cost Between Node and Base Station

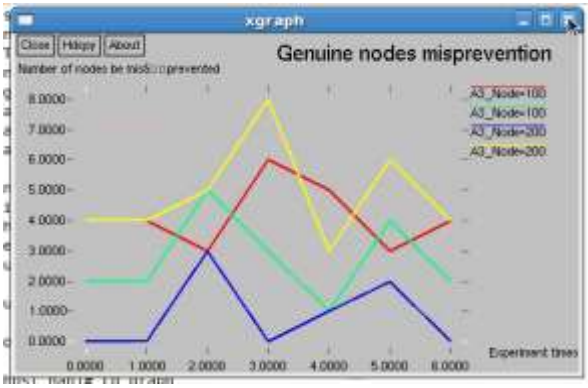
Method	Transmission size
With message recovery	$l_\sigma+l_{ID}+maxrg*l_{kID}$
Without message recovery	$l_\sigma+l_m$

From the data presented in figure 10, we can deduce that SRKD has a smaller communication overhead than QC. Please refer to the previous discussion regarding the assessment of the effectiveness of the storage for the explanation. The cost of communication has been shown to be significantly decreased thanks to the message recovery method, as is evident from both figure 10 and Table 5.5.

6 RESULTS AND ANALYSIS







CONCLUSIONS

We have come up with a brand new method of random key distribution that goes by the name of SRKD. This method of random key distribution is more resistant to information eavesdropping and the capture of nodes than existing methods of random key distribution are. There are two significant benefits that come with the SRKD that has been proposed. (1) It offers a protection technique that can be used to fend off the node replication attack. It is possible for SRKD to successfully enable the identification and revocation of replicas, and at the same time, it is possible for it to prevent replication nodes from injecting "fake information" to some level. (2) The SRKD protocol calls for a low overall energy consumption. Although the overhead for storage and communication in SRKD is lower than that of conventional QC, the cost of storage and communication in SRKD is slightly higher than that of QSP. Comparatively, the cost of storage and communication is lower than that of QSP. Nevertheless, we assert that this is the trade-off: an increase in a cost that is negligible in order to accomplish a higher level of security. In addition, we make advantage of the message recovery mechanism in order to cut down on the bandwidth cost that SRKD produces. In practise, SRKD may be able to assist us improve the security of WSS applications without severely compromising key bootstrapping and speed. This would be a big benefit. In our future work, we need to make improvements in a few different areas. For instance, we take into consideration efforts targeted at establishing a more suitable voting threshold for replication detection; optimising the upper bound s to ensure the connectivity of the network; and making the most of the opportunity to prevent attacks.

REFERENCES

1. G. Xu, Y. Cao, Y. Ren, X. Li, and Z. Feng, “Network security situation awareness based on semantic ontology and user-defined rules for internet of things,” IEEE Access, vol. 5, pp. 21 046–21 056, 2017.
2. H. Radhappa, L. Pan, J. Xi Zheng, and S. Wen, “Practical overview of security issues in wireless sensor network applications,” International Journal of Computers and Applications, vol. 40, no. 4, pp. 202–213, 2018.
3. L.Pan,X.Zheng,H.Chen,T.Luan,H.Bootwala,andL.Batten,“Cyber security attacks to modern vehicular systems,” Journal of Information Security and Applications, vol. 36, pp. 90–100, 2017.

4. X. Cao, J. Chen, Y. Xiao, and Y. Sun, "Building-environment control with wireless sensor and actuator networks: Centralized versus distributed," IEEE Transactions on Industrial Electronics, vol. 57, no. 11, pp. 3596–3605, 2010.
5. X. Zeng, G. Xu, Z. Xi, X. Yang, and W. Zhou, "E-aua: An efficient anonymous user authentication protocol for mobile iot," IEEE Internet of Things Journal, vol. PP, no. 99, pp. 1–1, 2018.
6. G. Xu, Y. Zhang, A. Sangaiah, X. Li, A. Castiglione, and X. Zheng, "Csp-e2: An abuse-free contract signing protocol with low-storage ttp for energy-efficient electronic transaction ecosystems," Information Sciences, vol. 476, pp. 505–515, 2019.
7. X. Liu, N. Xiong, N. Zhang, A. Liu, H. Shen, and C. Huang, "A trust with abstract information verified routing scheme for cyber-physical network," IEEE Access, vol. 6, pp. 3882–3898, 2018.
8. G. Xu, L. Jia, Y. Lu, X. Zeng, Z. Yao, and X. Li, "A novel efficient maka protocol with desynchronization for anonymous roaming service in global mobility networks," Journal of Network & Computer Applica- tions, vol. 107, p. S1084804518300407, 2018.
9. J. Ren, Y. Zhang, K. Zhang, A. Liu, J. Chen, and X. Shen, "Lifetime and energy hole evolution analysis in data-gathering wireless sensor networks," IEEE transactions on industrial informatics, vol. 12, no. 2, pp. 788–800, 2016.
10. J. Luo, J. Hu, D. Wu, and R. Li, "Opportunistic routing algorithm for relay node selection in wireless sensor networks," IEEE Transactions on Industrial Informatics, vol. 11, no. 1, pp. 112–121, 2015.
11. L. Oliveira, J. Rodrigues, A. deSousa, and V. Denisov, "Network admission control solution for 6lowpan networks based on symmetric key mechanisms," IEEE Transactions on Industrial Informatics, vol. 12, no. 6, pp. 2186–2195, 2016.
12. C.Yu,Y.Tsou,C.Lu,andS.Kuo,"Localizedalgorithmsfordetectionof node replication attacks in mobile sensor networks," IEEE Transactions on Information Forensics and Security, vol. 8, no. 5, pp. 754–768, 2013.
13. K. A. Shim, "Basis: A practical multi-user broadcast authentication scheme in wireless sensor networks," IEEE Transactions on Information Forensics and Security, vol. 12, no. 7, pp. 1545–1554, 2017.
14. L. Eschenauer and V. Gligor, "A key-management scheme for distributed sensor networks," in Proceedings of the 9th ACM Conference on Computer and Communications Security. ACM, 2002, pp. 41–47.
15. H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in IEEE Security and Privacy. IEEE, 2003, pp. 197–213.
16. F.Gandino,R.Ferrero,andM.Rebaudengo,"Akeydistributionscheme for mobile wireless sensor networks: q-s-composite," IEEE Transactions on Information Forensics and Security, vol. 12, no. 1, pp. 34–47, 2017.
17. B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in IEEE Security and Privacy. IEEE, 2005, pp. 49–63.
18. B. Zhu, S. Setia, S. Jajodia, S. Roy, and L. Wang, "Localized multicast: Efficient and distributed replica detection in large-scale sensor networks," IEEE Transactions on Mobile Computing, vol. 9, no. 7, pp. 913–926, 2010.
19. M.Conti,R.DiPietro,L.Mancini,andA.Mei,"Arandomized,efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks," in Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing. ACM, 2007, pp. 80–89.
20. K. Xing and X. Cheng, "From time domain to space domain: Detecting replica attacks in mobile ad hoc networks," in IEEE INFOCOM. IEEE, 2010, pp. 1–9.
21. ashdawn,"Srkd,"https://github.com/ashdawn/SRKD, accessed Decem- ber, 2018.
22. AVISPA, "Automated validation of internet security protocols and applications," http://www.avispa-project.org/,accessed December,2018