
A. Ali¹, N. Fisal²
Faculty of Electrical Engineering,
University Technology Malaysia,
81310 Johor Bahru, Johor Darul Ta’zim, Malaysia
Email: ¹adelali3@lycos.com, ²sheila@suria.fke.utm.my

Abstract—Wireless sensor network is a wireless ad hoc network that consists of very large number of tiny sensor nodes communicating with each other with limited power and memory constrain. WSN demands real-time routing which requires messages to be delivered within their end-to-end deadlines (packet lifetime). Since many sensor networks will be deployed in critical applications, security is essential. Recently, many real-time routing protocols have been proposed, but none is designed with security as a goal. This paper proposes a novel secure enhancement for real-time routing protocol that provides secure real-time data in WSN. The proposed security countermeasures HELLO flooding and selective forwarding attacks. It ensures high packet throughput and minimized power consumption in the present of adversary nodes. The proposed security has been successfully studied through simulation work.

I. INTRODUCTION

The recent technological advancement in wireless communications, micro-electro-mechanical systems (MEMS), and digital electronics have led to the development of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate within short distances [1]. As can be shown in Fig. 1, the tiny sensor node consists of sensing, data processing, and communicating components. The sensor nodes can be interconnected to form a network defined as wireless sensor network (WSN). WSN consists of very large number of sensor nodes which are densely deployed either inside an event area or in proximity. WSN enables reliable monitoring and analysis of a physical environment.

Real-time communication is necessary in many WSN applications. For example, in a fire fighting application, appropriate actions should be made in the event area immediately as delay may cause huge damages further. The sensor data collected and delivered must still be valid at the time of decision making since late delivery of data may endanger the fire fighter’s life.

The general research challenges for multi-hop routing in WSN arise primarily due to the large number of constraints that must be simultaneously satisfied. One of the most important constraints on sensor nodes is the low power consumption requirement. Sensor nodes carry limited, generally irreplaceable power sources. WSN applications must operate for months or years without wired power supplies and battery replaced or recharged. Therefore, the power consumption must be considered while designing multi-hop routing in order to prolong the WSN lifetime [2].

Fig. 1. WSN architecture with MICAZ motes

Since many WSNs will be deployed in critical applications, security is essential. Unfortunately, security may be the most difficult problem to solve in WSNs [3, 4]. In particular, it is easy to eavesdrop or cause a network layer attacks which fall into one of the following categories: manipulating routing information, selective forwarding, Sybil, sinkhole, wormhole, and Hello flooding (unidirectional) attacks. Furthermore, most real-time communication and coordination routing protocols do not address security, so it is easy for an adversary to exploit those routing protocols on a given WSNs [5, 6]. Lightweight security schemes are required in real-time routing protocols for WSN. The security for real-time routing protocols must exploit the nature of the sensor network and relate to issues such as most data is only valid for a short time [3, 5]. Real-time routing protocol protocols designed for WSN must therefore balance real-time performance, energy efficiency and data security.

This paper presents secure real-time with load distribution SRTLD routing protocol that depends on optimal forwarding (OF) decision that takes into account of the link quality, packet delay time and the remaining power of next hop sensor nodes. It also presents the security enhancement which uses the encryption and decryption with authentication of the packet header to supplement secure packet transfer.
The remaining parts of this paper are organized as follows: Section II will present related work. The overview of SRTLD routing protocol will be described in Section III and Section IV will describe the security enhancement for SRTLD routing protocol. Section V will present the simulation study of SRTLD. Finally Section VI will conclude the paper.

II. RELATED WORK

Several solutions for securing WSNs have been proposed in the literature. TinySec by [7] presents two different security options: authenticated encryption (TinySec-AE) and authentication only (TinySec-Auth). In TinySec-AE, TinySec encrypts the data payload and authenticates the packet with a message authentication code. The message authentication code is computed over the encrypted data and the packet header. In TinySec-Auth, TinySec authenticates the entire packet with a message authentication code, but the data payload is not encrypted. TinySec currently runs on MICA platforms. TinySec-AE and TinySec-Auth increase packet latencies (compared to the current TinyOS stack) by 56.6 ms and 53.4 ms respectively. TinySec does not provide a secure localisation, secure routing mechanism while the proposed security algorithm addresses these weaknesses. It requires 728 bytes of RAM and 7146 bytes of program space. TinyHash based on hash algorithm for WSNs is a modified security method for TinySec [8]. It implements security components for message hash and authentication using Secure Hash Algorithm 1 (SHA1) function instead of using Cipher Block Chaining-Message Authentication Code (CBC-MAC) based on SkipJack, which is offered in TinySec. TinyHash uses Hash Message Authentication Code (HMAC) scheme for authentication and SHA1 hash algorithm for message digest. It is implemented on Telos mote using SHA1 8bits version in order to have the compatibility with MICA mote series. SHA1 requires 140 bytes of RAM and 3504 bytes ROM. Also, it requires 35ms execution time for 160 bits data packet. HMAC has twice the size of SHA1 algorithm and has twice the execution time of SHA1.

One-time pad was invented by [9]. It is a very simple security system and is unbreakable [10]. The pad is a block of random data equal in length to the original message and one copy of the pad is kept by each user. The word random is used in its most literal possible sense here. If the data on the pad is not truly random, the security of the pad is reduced. The pad is used by XORing every bit of the pad with every bit of the original message. Once the message is encoded with the pad, the pad is destroyed and the encoded message is sent. On the recipient side, the encoded message is XORed with the duplicate copy of the pad and the plaintext message is generated. The drawbacks of this mechanism are producing real random number is complicated and a one-time pad does not provide data authenticity [10]. Furthermore, if an adversary captures and stores an encrypted message $C = M \oplus X$, where $X$ is the one-time pad, and later on gets $X$ (steals the code book), then he can decode: $M = X \oplus C$.

The same holds for any existing private or public key encryption scheme.

The existing security methods are not applicable for real-time routing in WSNs because the execution time for one-hop is high and WSNs have density deployment where hundreds of nodes need more time to process security mechanism.

III. SRTLD ROUTING PROTOCOL OVERVIEW

In Fig. 2, SRTLD consists of five functional modules that include location management, routing management, power management, neighborhood management and security. The location management in each sensor node calculates its location based on the distance to three pre-determined neighbor nodes. The power management determines the state of transceiver power and the transmission power of the sensor node. The neighborhood management discovers a subset of forwarding candidate nodes and maintains a neighbor table of the forwarding candidate nodes. The routing management computes the optimal forwarding choice, makes forwarding decision and implements routing problem handler. The security management applies encryption and decryption with authentication mechanisms at specific fields in the packet header. [11] describes SRTLD in more details.

In order to carry out the optimal forwarding calculation, the routing management calculates three parameters that include packet velocity, remaining power and link quality. The wireless link quality at the physical layer is studied to predict the communication between sensors. In addition, the remaining power is estimated to spread all traffic load distribution during path forwarding to the destination. Eventually, the router management will forward a data packet to the one-hop neighbor that has an optimal forwarding. The optimal forwarding (OF) is computed as follows:

$$OF = \lambda_1 \cdot PRR + \lambda_2 \cdot \frac{V_{\text{batt}}}{V_m} + \lambda_3 \cdot \frac{V}{V_m}$$

Where: $\lambda_1 + \lambda_2 + \lambda_3 = 1$ \hspace{1cm} (1)

where $V_{\text{batt}}$ is the maximum battery voltage for sensor nodes and is equal to 3.6 volts [12]. $V_m$ is the maximum velocity of the RF signal that is equal to the speed of light. The determination of PRR, $V_{\text{batt}}$ and $V$ is elaborated in the following section. The values of $\lambda_1$, $\lambda_2$ and $\lambda_3$ are estimated by exhaustive search using Network Simulator-2 (NS-2) simulation such that $\lambda_1 + \lambda_2 + \lambda_3 = 1$ as illustrated in [11]. In [11], the number of possible values for each $\lambda$ is 11 (from 0.0 to 1.0) and the number of tails for event $\lambda_1 + \lambda_2 + \lambda_3 = 1$ is 66. The optimal trail from the 66 trails has been determined using NS-2 simulation with four types of grid network topology which are low density, medium density, high density with one traffic source and high density with several traffic sources. The finding in [11] shows that the trail with 0.6, 0.2 and 0.2 for $\lambda_1$, $\lambda_2$ and $\lambda_3$ experiences high performance in term of delivery ratio and power consumption. Therefore, equation 1 can be written as;
The enhanced security in SRTLD is based on work done on One-time Pad [9]. The modification solves the weaknesses of [9] such as producing real random number and data authenticity. Real-time routing protocols are limited to time constrain which is an important parameter to consider when designing security in SRTLD routing protocol. Since sensor nodes function as routers, the encryption and decryption with authentication process should be made at every hop for every forwarding packet in WSN. Thus, the security enhancement in SRTLD must ensure real-time routing between the source and the destination. In addition, SRTLD solves the problem of producing real random number problem in [9] using random generator function encrypted with mathematical function. The output of random function is used to encrypt specific header fields in the packet such as source, destination addresses and packet ID. Moreover, the data authenticity problem in [9] is solved in SRTLD using authentication procedure applied after decryption. These are the main differences between securities in SRTLD and [9]. The output packet of the security system (secure packet) is a packet with incorrect header fields. If an adversary node eavesdrops the secure packet, it does not know the source, the destination and the packet ID of the secure packet. Since the secure packet is only valid for a short time, the dynamic mathematical calculation for decryption is designed to prevent an adversary to understand the secure packet as explained in the next chapter. The proposed security mechanism is designed based on the following assumptions:

- Each sensor node is static and aware of its location.
- Sink is a trusted computing base.
- Pseudo random function as a function of master key and packet ID (Pkt_ID) is stored during program uploading into sensor nodes.
- Hard mathematical function with its reverse calculation is stored in each sensor node before sensor deployment.
- Two master keys (k, kl) are stored during program uploading into sensor nodes. Where k is used as a master key in all nodes for encryption and decryption purpose and kl is used as a master key for a new node after WSN is already started.

In order to satisfy the producing real random number requirement, the security management uses the random generator function \( F_i(Pkt_ID) \) recommended by [13] as:

\[
F_i(Pkt_{ID}+1) = A \cdot F_i(Pkt_{ID}) \mod P
\]

where \( P \) is a prime and equal \((2^{31}-1)\) for a 32-bit (31 bit + 1 sign bit), \( A \) is a positive primitive root of \( P \). Any power (modulo \( P \)) of \( 7 \), say \( 7^k \) where \( k \) is not a factor of \( P - 1 \), is also a positive primitive root of \( P \). In this way, many \( A \)'s could be generated, and it was confirmed empirically that an \( A \) approximately equal to \((P)^{1/3}\) is required to even begin to give good test results. Hence, \( k \) is 5 and \( A \) is set to \( 7^5 \) for \((2^{31}-1)\). The randomness of equation 3 has been tested and analyzed in [13]. The random function \( F_i(Pkt_{ID}) \) will create random number based on the \( k \) and the \( Pkt_{ID} \). Then, the mathematical function \( f(x) \) is applied as:

\[
f(x) = (f(x + F_i(Pkt_{ID}))^1 + R) \mod (2^{31}-1)
\]

where \( R \) is number of sensor devices in the same group of WSN and \( x \) is S_ID or D_ID. Finally, the encrypted packet (ciphertext) will be sent.

Since the legal received node has same random generator, it can create the same random number to decrypt the original packet based on \( k \) and Pkt_ID. Therefore, the reverse function of the security management in the previous example can be computed as follows:

\[
x = ((f(x) - R)^1 - F_i(Pkt_{ID})
\]

The authentication procedure will check the output of decryption procedure. If the output of decryption procedure is between 0 and \( R \), the authenticity status is ok. Otherwise, the authenticity status is in error. It is interesting to note that the encryption mathematical function in this proposal is an example, however in reality; the encryption mathematical function is selected based on harder reversing mathematical function. It is also important to note that new sensor device can be added to the WSN using a control packet. In this control packet, the new \( R \) is encrypted by the same encryption mechanism with \( kl \) as a master key.

V. SIMULATION STUDY

The simulation evaluates the capability of proposed security in SRTLD to overcome the HELLO flood and selective forwarding attacks. A realistic simulation environment for SRTLD was created based on the physical characteristics of MICAZ sensor node [12]. Many-to-one traffic pattern is used. In this work, 121 nodes are distributed in a 100m x 100m region as shown in Fig. 3. Nodes numbered as 120, 110, 100 and 90 are the source nodes, node 0 is the sink and nodes numbered as 24, 25, 31 and 36 are adversary nodes. Table 1 shows the simulation parameters of SRTLD routing protocol in NS-2. In this simulations study, the packet rates were varied while the end-to-end deadline
and simulation time were fixed at 250 ms and 100s respectively.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>IEEE 802.15.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propagation Model</td>
<td>Shadowing</td>
</tr>
<tr>
<td>path loss exponent</td>
<td>2.5</td>
</tr>
<tr>
<td>shadowing deviation</td>
<td>4.0</td>
</tr>
<tr>
<td>reference distance (m)</td>
<td>1.0</td>
</tr>
<tr>
<td>phyType</td>
<td>Phy/WirelessPhy/802_15_4</td>
</tr>
<tr>
<td>macType</td>
<td>Mac/802_15_4</td>
</tr>
<tr>
<td>Initial Energy</td>
<td>3.3 Joule</td>
</tr>
<tr>
<td>Power transmission</td>
<td>1 mW</td>
</tr>
<tr>
<td>Traffic</td>
<td>Constant Bit Rate (CBR)</td>
</tr>
</tbody>
</table>

A. Attacks on WSNs Routing

Many WSNs routing protocols are quite simple, and for this reason are sometimes even more susceptible to attacks against general ad-hoc routing protocols. In WSN, an adversary can either deploy his own node or compromise some nodes. The manipulated sensor data attacks are divided into two classes that include manipulating user data directly and influencing the underlying routing topology. SRTLD defends against the attacks that influence the underlying routing protocol such as selective forwarding, sinkhole, Sybil, wormholes, and HELLO flood attacks [5, 6].

♦ Selective Forwarding

In a selective forwarding attack, malicious nodes may refuse to forward certain messages and simply drop them [5]. The neighboring nodes will conclude that the current route has failed and they decided to seek another route. Selective forwarding attacks are typically most effective when the attacker is explicitly included on the path of a data flow. However, it is conceivable that an adversary overhearing a flow passing through neighboring nodes might be able to emulate selective forwarding by jamming or causing a collision on each forwarded packet of interest. Thus, an adversary who is launching a selective forwarding attack will follow the path of least resistance and attempt to include herself on the path of the data flow.

♦ HELLO Flood Attack

Many routing protocols require nodes to broadcast HELLO packets to announce themselves to their neighbors, and a node receiving such a packet may assume that it is within radio range of the sender [5]. This assumption may be false; a laptop-class attacker which is broadcasting routing or other information with large enough transmission power, could convince every node in the network that the adversary is its neighbor. For example, an adversary who is advertising a very high quality route to the base station to every node in the network could cause a large number of nodes to attempt to use this route, but those nodes sufficiently far away from the adversary would be sending packets into oblivion. The network is left in a state of confusion. A node realizing the link to the adversary is false could be left with few options: all its neighbors might be attempting to forward packets to the adversary as well.

A. Influence of HELLO Flood and Selective Forwarding Attacks

In order to analyze the security performance in SRTLD, delivery ratio, packet overhead and power consumption are studied in the presence of HELLO flood and selective forwarding attacks. SRTLD routing that uses security enhancement is defined as (SRTLD_s). Fig. 4(a) shows that SRTLD experiences higher delivery ratio by 6.83 % than SRTLD_s when there are no attacks. This is because SRTLD_s requires 4.2 ms additional processing delay to implement the security mechanism for one-hop as illustrated in Fig. 4(b). The total processing delay due to security mechanism affects the packet deadline, which results in some data packets missing the end-to-end deadline. However, the delivery ratio of SRTLD is reduced by 50 % when the adversary node injects HELLO flood and selective forward attacks as shown in Fig. 4(c). This is primarily because SRTLD_s cannot defend against the HELLO flood and selective forwarding attacks. The simulation results in Fig. 5(a) show that SRTLD without attack consumes less power than SRTLD_s because it spends fewer packets overhead. However, Fig. 5(b) shows that the normalized power consumption in SRTLD under attack is higher than normalized power consumption in SRTLD_s. The reason is due to the packet dropping caused by selective forwarding attack in SRTLD.
VI. CONCLUSION

Most of existing security mechanisms designed for WSN do not consider the real-time routing and the real-time routing protocol have been proposed none is designed with security as a goal. This paper presents security for SRTLD routing protocol in WSN. The execution time factor for implementing security in WSN is very important because it will cause extra delay for the packet when it is forwarded through multi-hop neighbors. The finding concludes that the security mechanism requires only small memory size about 216 bytes. It experienced minimum execution time of 4.2 ms for one hop which is much lesser than the existing security mechanisms in WSN.

ACKNOWLEDGMENT

The authors wish to thank the Ministry of Science, Technology and Innovation (MOSTI) for funding this research. Our gratitude also goes to the University Technology Malaysia (UTM) for financial support.

REFERENCES


