http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 8 (5): 678-687, 2009
ISSN 1812-5638

DOSM: A Data-Oriented Security Model Based on Information Hiding in WSNs

[1]Xiangrong Xiao, [1]Xingming Sun, [2]Xinbing Wang and [3]Lei Rao
[1]School of Computer and Communication, Hunan University, Changsha, 410082, China
[2]Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai, 200240, China
[3]School of Computer Science, McGill University, Montreal, Quebec, H3A2A7, Canada

Abstract: In this study, we propose a non-cryptology and protocol-independent technique based on information hiding, called Data-Oriented Security Model (DOSM). Instead of one layer of protection, the proposed scheme offers two-fold protection against attack. An attacker first explores whether the data in question carries any useful information and then conducts extraction on it. Information is concealed by changing some properties of the data, which does not incur extra overheads to the sensor nodes. With the help of DOSM, each forwarding node can verify the embedded marks using the source node ID in each packet. This mechanism is used to filter data in a distributed manner. It can also avoid fake and tamper attacks by terminating bad packets as soon as they are detected. The data security is guaranteed by filtering the inconsistent packets between the hidden data and its source ID, which yields low energy consumption and high reliability. The simulation results show that the proposed DOSM protects the security of data communication and achieves data authentication invisibly at small overhead expense.

Key words: Information hiding, identity authorization, protocol-independent, data filtering, low consumption

INTRODUCTION

The rapid development of Wireless Sensor Networks (WSNs) brings pervasive computation to many applications. Consequently, attackers have more incentive to take control of one or more sensors and launch active attacks to break networks, such as eavesdropping, intercepting, injecting and tampering. WSNs need to guarantee the security of transmitted data, which is the most important issue including confidentiality, authentication, integrity and freshness (Deb *et al*., 2003). These problems are mostly solved by cryptography or security protocols, but rarely with information hiding technology. The existing security techniques are inadequate. We are working on seeking novel and more effective measures to prevent communication vulnerabilities being exposed by any means.

One of the primary challenges in the security design is energy constraint. Thus, it is improper to use those algorithms, including public key algorithm and digital signature algorithm, which are high in power consumption (Akyildiz *et al*., 2002). Due to memory and computation limitations, they are considered less secure than AES. Liu and Ning (2008) developed TinyECC to provide flexible integration of ECC-based PKC (Public Key Cryptography) in sensor network applications. Bremler-Barr and Levy (2005) proposed SPM which adds a key to each packet to validate whether the packet was spoofed. This scheme induces heavy traffic for they need to send tables of 120 kb. Moreover, the encrypted messages become the explicit targets of attackers.

Similar to the cryptography approaches, security protocols require special configurations and increase traffic with extra space in packets. For example, TinySec (Karlof *et al*., 2004) and SPINS (SNEP, μTESLA) (Perrig *et al*., 2002) are link layer security solutions, which add 8 bytes per message. MiniSec (Luk *et al*., 2007) is a network layer security protocol with MiniSec-U for unicast and MiniSec-B for broadcast. It adds 13 bytes of extra data to each packet which can only be used for special motes. An approach is developed (Ning *et al*., 2008) to mitigate the DoS attacks against broadcast authentication. This approach requires significant computational power at the sender. These protocols rely exclusively on software-level encryption and authenticating routines.

Information hiding techniques have been established over the past few decades and aim to hide the existence of messages. It is a new research area with the main objectives being anti-fraud, protection of intellectual property and covert communication. Information hiding makes useful data invisible instead of unreadable, which avoids making the cryptographs attractive to attackers and resolves the unmanageable problem of cryptographs

Corresponding Author: Xingming Sun, School of Computer and Communication, Hunan University, Changsha, 410082, China
Tel: 86-731-8821341 Fax: 86-731-8821780

going beyond decryption. Recently, Feng and Potkonjak (2003) propose a real-time watermarking technique by embedding authorship signatures into sensing data, which imposed discrepancies between the expected distance and the actual measurements. Smith *et al*. (2005) have embedded a bit stream into an RFID time-series channel by ID modulation and used it as node authentication. The robust watermarking (Mitrea *et al*., 2008) is proposed for protecting streaming video against piracy. Kleider *et al*. (2004) hid marks to OFDM in the physical layer by baseband waveform modulation. These approaches can be treated as physical layer secure accesses that need special setup equipment. Sion *et al*. (2006) and Guo *et al*. (2007) treat stock prices, environmental sensing and intrusion detection of network as numeric streams. Those methods can only be equipped in the data center without capability limitations.

Compared with attackers, sensor resources are so limited that most sensor-acceptable algorithms are not computationally strong enough. On the other hand, the challenge is that packet storage space is limited and therefore increasing its size in existing security protocols is not suitable because they need extra traffic overload. Neither can they provide end-to-end security guarantees, particularly with a large number of nodes which makes WSNs vulnerable to different attacks. Information hiding is a new solution for streaming data protection in WSNs. It provides a novel way that is also effective. The appropriate security mechanism, which has a tradeoff between the computation cost and security intensity, should be data-oriented and application-based.

Compared to existing secure mechanisms, we propose the Data-Oriented Security Model (DOSM) based on the Information Hiding (IH) technique used to address the above challenges. Through this technique, the embedding process predefines a mapping between the hidden data and the cover-message (Petitcolas *et al*., 1999) which is only known by the owner. Different from encryption which makes the data unreadable, IH hides their very existence. Generally, the least significant properties of data that do not determine the meaningful content of the original message are replaced with new values in a contracted way that causes the least amount of distortion. Hence, the length of a cover-message does not change during embedding. The cover-message containing hidden data is nonexistent as far as meaningful content is concerned.

Through, the above method, the source node conceals its ID in the data before sending packets to intermediate nodes. Intermediate nodes verify the hidden marks with the source ID before forwarding packets. Intermediate nodes relay the packets if they are consistent; otherwise, intermediate nodes terminate the forwarding process. The algorithm can avoid transmitting useless messages and distinguish malicious nodes from ordinary ones without any global information.

The simulation results show that DOSM is robust against many attacks such as fakes, spoofing and Sybil. The contributions of this study are as follows:

- The DOSM model protects end-to-end data in every forwarded packet. The data security is guaranteed only by filtering the inconsistent packets between the hidden data and its source ID, which yields low energy consumption and high reliability
- We design embedding and decoding algorithms for sensor nodes, which is compatible with other security protocols in a transparent manner and is protocol-independent. There are three data security algorithms for different situations: initial, compact and fragile

PROBLEM STATEMENT

Sensors can gather messages such as temperature, humidity, illumination or other sensitive data. For example, a company may attempt to prevent sensors from collecting and transferring environment pollution data. Preventive measures need to be more stringent in battlefield monitoring to acquire military data such as the whereabouts of weapons. Security mechanisms should be robust to defend themselves against various attacks and avoid vulnerable exposure, Hence; we propose DOSM based IH techniques to avoid wasting energy during packet transmission.

Traditionally, intermediate nodes might relay useless packets that waste energy. Adopting DOSM, every credible node acquires data as normal and embeds its ID, combined with its private message (e.g., message count, data digester), into the packet. Intermediate nodes extract the hidden mark and check the source ID. Only consistent packets are forwarded to the next node. Otherwise, the packet is discarded and the transmission is terminated.

To make the motivation scenario more readily understandable, we show through an example the transmission scenarios. As shown in Fig. 1, sensor nodes are deployed in the sensing area where the credible nodes are denoted as white and the malicious ones are black. There are two different scenarios.

Secure scenario: Node 1 No. acquires data and firstly embeds its ID into the packet. Then the intermediate node 5 No. forwards the packet to 6 No. after it passes verification. Nodes 9, 10 and 11 No. do the same in turn.

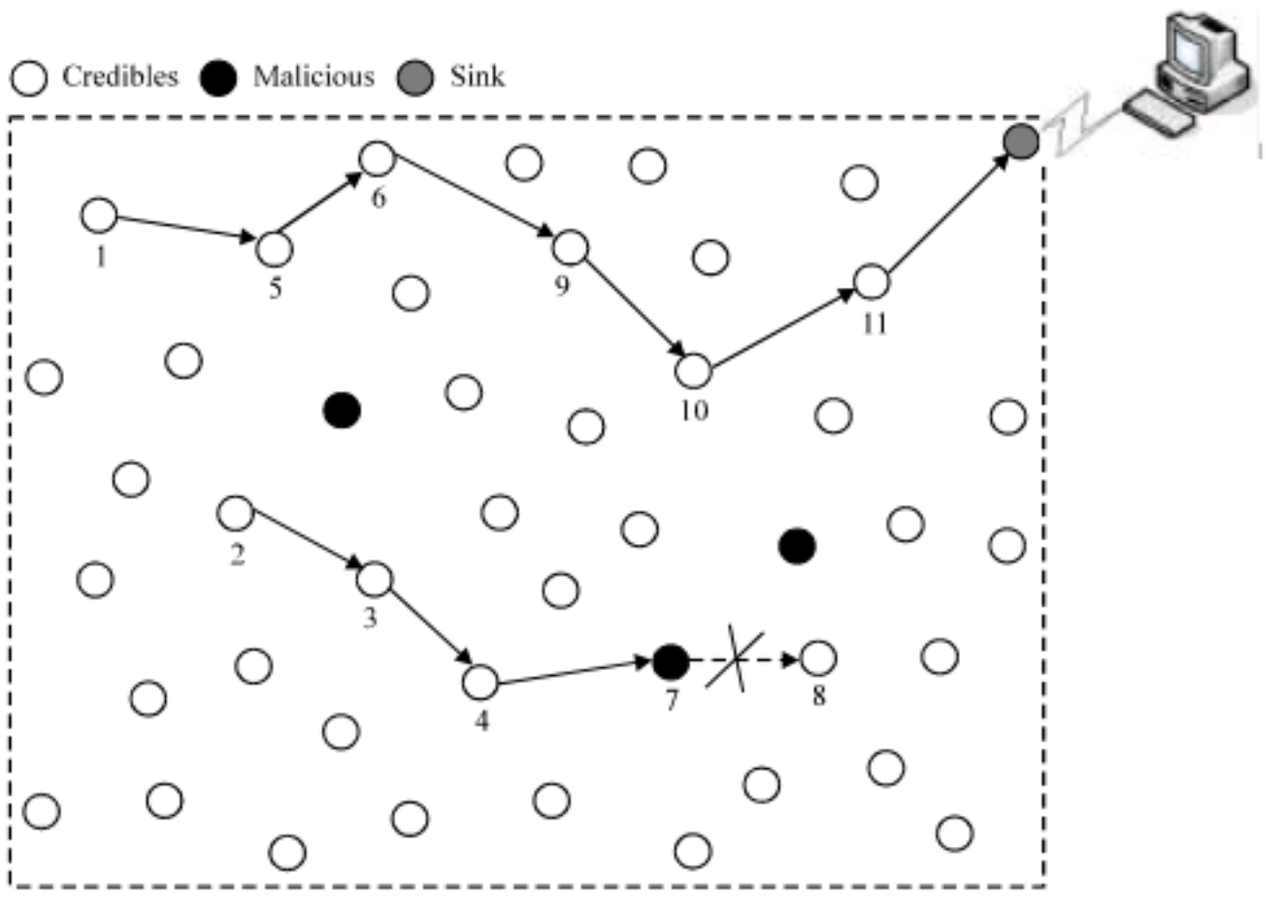

Fig. 1: Threat Scenario and Secure Relay

At last, the sink obtains the authentic data. The up LOC (Lines of Communication) in Fig. 1 shows this scenario.

Attack scenario: Node 2 No. acquires data and embeds its ID into packet as above. The packet is sent to nodes 3, 4 and 7 No. Unfortunately, node 7 No. is malicious. After that the packet might be tampered with, faked or replayed. Node 8 No. terminates the inconsistency packet by checking in DOSM, which avoids wasting energy and immune from injecting. In that situation, the falsified data would be filtered and the adversary 7 No. would become distinguishable. The lower LOC in Fig. 1 shows this scenario.

We now present some assumptions that will be used in the study. We assume that an adversary does not have the capability to attack the Base Station because it is well protected. The attacker might interfere with the communication of nodes or eavesdrop on the radio channel.

THE DOSM MODEL

DOSM is deployed in TinyOs by modifying its Active Message (AM). We do not change other fields like the FCF (frame control field) or the like. The packet format is shown as follows (Buonadonna *et al*., 2001).

DEST(2)	AM(1)	Len(1)	Grp(1)	Data(20)	CRC(2)

There are 7 bytes of additional overheads in each packet. Metadata (payload), which occupies 20 bytes, is placed in the Data field. In this experiments, each datum occupies 2 bytes, hence each packet contains 10 data denoted as d_i (i = 1,…,10).

Before designing the detailed embedding and decoding algorithm, we first define base-bit and embedded-bit for each datum.

Definition 1: Base-bit is the poison bit in a datum for embedding reference, denoted as b.

Definition 2: Embedded-bit carries information in a datum, denoted as e.

We can obtain a base-bit and embedded-bit in every datum of packets. Then we explore the embedding and decoding algorithms by mapping the switching state between them.

Figure 2 shows the DOSM mechanism that describes the transmission process.

DOSM has three roles:

- Sense node
- Relay node
- Sink node

A sense node can also be used as a relay node in other communications according to their allocated tasks. In the sense node, the node ID is embedded into the packet before transmission. The relay node will verify the forwarding packets by checking the consistency of the source ID and the extracted marks. Only authentic packets are transmitted to the next node. Through, the packet filtering, the nodes in an entire network can save energy for terminating suspicious transmissions.

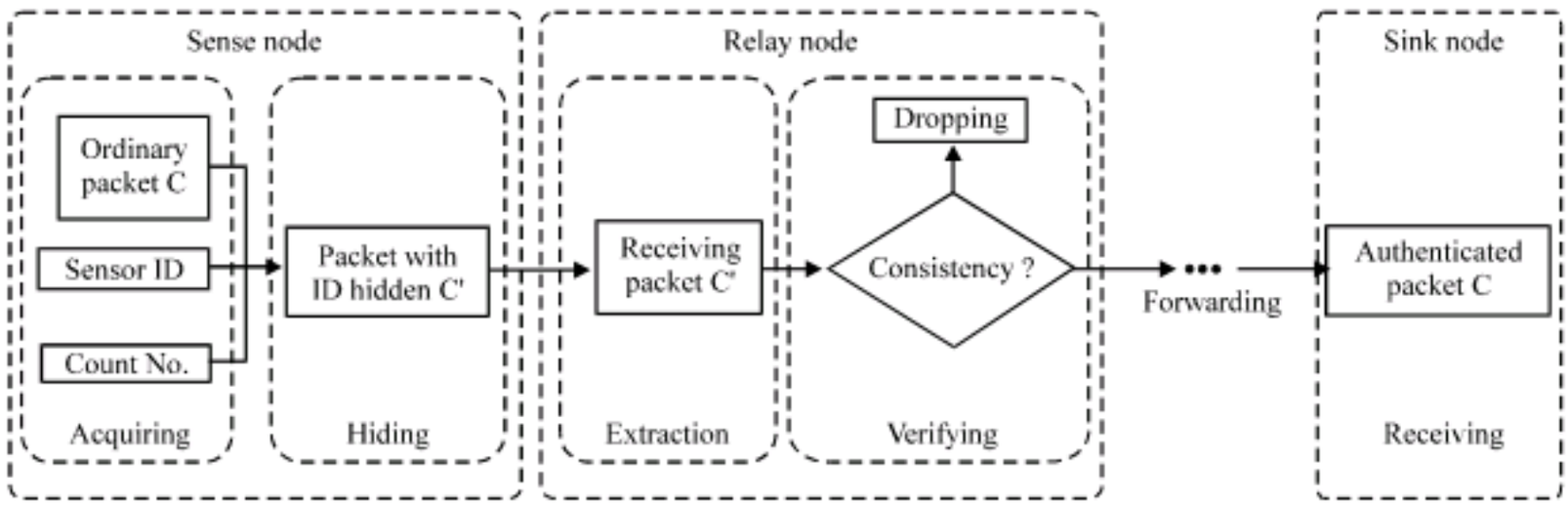

Fig. 2: DOSM Implementation

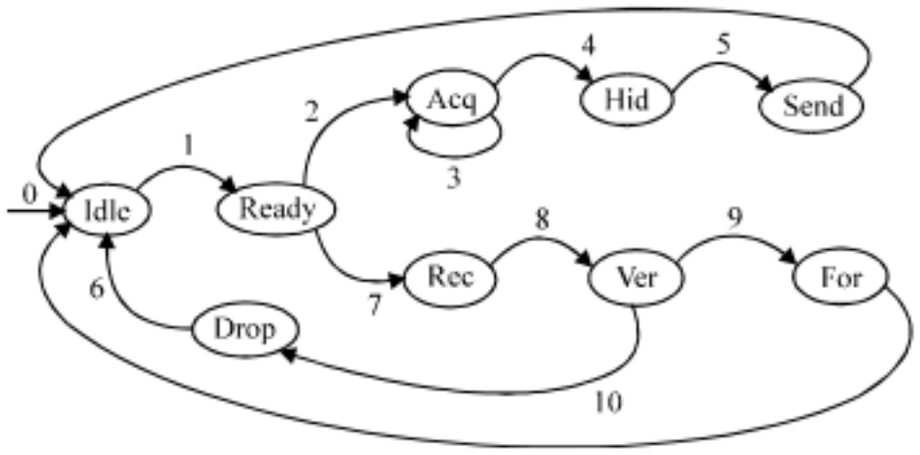

Fig. 3: State transition diagram, 0: Initialization, 1: Wake up (acquiring/receiving), 2: Begin to acquire, 3: Acquiring data, 4: Hiding data, 5: Sending packet after hiding, 6: Back to idle, 7: Begin to receive, 8: Identify verification, 9: Forwarding authentic packet and 10: Dropping unauthentic packet

There are 9 possible states: IDLE, READY, ACQ, HID, SEND, REC, VER, FOR and DROP, none of which makes the transition to other states until specific conditions are satisfied. Figure 3 is the state transition diagram at the sending end.

Nodes do not attempt to acquire or transmit in the IDLE state. They simply check whether there is any data to acquire or packets to transmit. When a node has data to acquire, it moves to ACQ and starts to acquire data until the packet is filled. Then the node moves to HID for hiding and SEND to transmit. The node moves to REC for receiving when it is woken up to transmit and VER to verify its identity. Next it moves to the FOR state to forward the packet to the next node if it passes the identification stage. Otherwise, it moves to DROP for dropping the suspicious packet. After a successful SEND, FOR or DROP, the node goes backs to IDLE.

Basic design: DOSM includes two algorithms especially for sensors: embedding and decoding/verifying. The embedding algorithm hides the identifier tag in the cover-message; the decoding/verifying algorithm extracts the mark from the carrier and checks the consistency. Through these approaches, intermediate nodes filter the tampered or forged packets to avoid wasting energy.

Embedding algorithm: The sender embeds the node-ID combined with some private digesters. We set a mapping denoting the hidden bits between the embedded-bit and the base-bit. This mapping is described in rule 1 of the embedding process.

Rule 1: (Embedding): We embed one bit h into a datum by changing the value of the embedded-bit when comparing it to the base-bit.

h = 1: Make switching, i.e., set embedded-bit = 1 if base-bit = 0 or embedded-bit = 0 if base-bit = 1;
h = 0: Make no switching, i.e., set embedded-bit = '1' if base-bit = 1 or embedded-bit = 0 if base-bit = 0.

We design this switching state mapping as a contract for expressing the hidden bit. The algorithm for embedding is shown in Algorithm 1.

ALGORITHM 1

Embedding:
Input: original packet C, ID, count N, key K
Output: sending packet with ID hidden C'

Body:

```
1:   h = Encrypt (ID, K)
2:   for i = 0 to len(h)-1
3:      bp = Hash (K, N, dit) mod 15
4:      b = data[bp, i]
5:         if (h[i] = 1 and b = 1) then
6:                  data [15, i] = 0    //make switching
7:                  else if (h[i] = 0 and b = 1) then
8:                  data [15, i] = 0   //make no switching
```

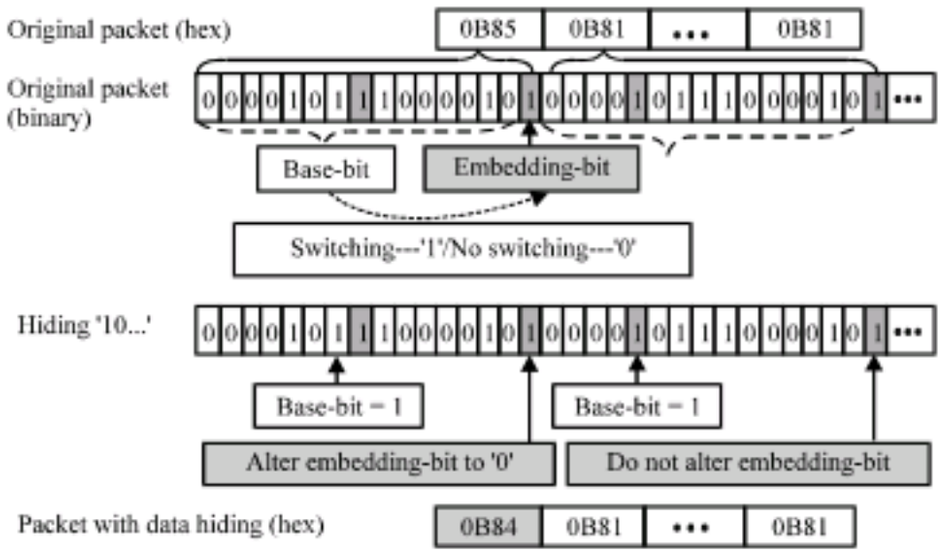

Fig. 4: Data hiding with basic algorithm

```
9:                endif
10:           endif
11:       endfor
12:   update C
13:   Return C'
```

First, the node ID is encrypted using a simple symmetrical cipher in line 1. Then it is embedded into the packet in lines 2~11. We compute the base-bit position by the Hash digester with parameters K, N and d_{it} in line 3, where d_{it} is the first t bits of data d_i. The first 15 bits of a datum are used for the digester, while, the last one is not included as it might change during the embedding. The algorithm computes the base-bit in line 4, then makes there has switching or not by changing the embedded-bit by comparing it with the base-bit according to h in lines 5~8. Finally, the data are updated with the new values.

Figure 4 is an example, in which the embedded-bit is the last bit of a datum in our basic solution. The base-bit position relates to key, count and the data value that might change in a different hiding process. We obtain base-bit = 1 in the first datum and alter the embedded-bit to '0' when hiding '1'; we also obtain base-bit = 1 in the second datum, without altering the embedded-bit in hiding '0'.

Decoding and verifying: Intermediate nodes decode the received packet and verify the extracted marks with the source node ID before forwarding. The mapping switch between the base-bit and embedded-bit represents bit '1' or '0' as defined in Rule1. Decoding is a converse process to embedding, which is shown in rule 2. Likewise, we obtain '1' when switching and '0' for no switching.

Rule 2: (Decoding): We extract the hidden bit h by comparing the switching state between the base-bit and embedded-bit.

h = 1: Switching, i.e., if base-bit = '0' and embedded-bit = '1'; or base-bit = '1' and embedded-bit = '0';
h = 0: No switching, i.e., if base-bit = '0' and embedded-bit = '0'; or base-bit = '1' and embedded-bit = '1'.

The decoding and verifying algorithm are listed in Algorithm 2.

ALGORITHM 2

Decoding and verifying:
Input: receiving packet C'', ID, count N, key K
Output: verified result R

Body:

```
1:   for i = 0 to 9
2:       bp = Hash (K, N, dit) mod 15
3:       b = data[bp, i]          //obtain base-bit
4:       e = data[15, i]          // obtain embedded-bit
5:       if (b = e) then
6:           h[i] = 0      //no switching----0
7:       else
8:           h[i] = 1      //switching----1
9:       endif
10:   endfor
11:  if (decrypt (h, K) = ID) then R = true
12:      else R = false
13:  endif
14:  Return R
```

We obtain a base-bit and an embedded-bit similar to the embedding and check the appointed mapping for switching the state of every datum in the packet in lines 2~4. We extract the hidden information h according to rule 2 in lines 5~9. Following the extraction, we decrypt h by K and verify the consistency. The decoded mark is different to the source ID meaning that the identity verification passes, so the intermediate nodes would consider terminating the relaying. Therefore, we can distinguish the forged data and the malicious nodes as soon as possible.

Precision and security: We embed the source ID into acquired data by Telos mote (Ford, 2007) and use its packet with 10 data contained. Figure 5 compares data values before and after hiding the mark.

Through comparison, we find that the hidden mark changes the data value slightly. Since, the acquired data are inaccurate, errors within a small range are acceptable. Slight distortions introduced by data hiding are tolerable. Figure 6 is the data precision and security with a different hiding capacity.

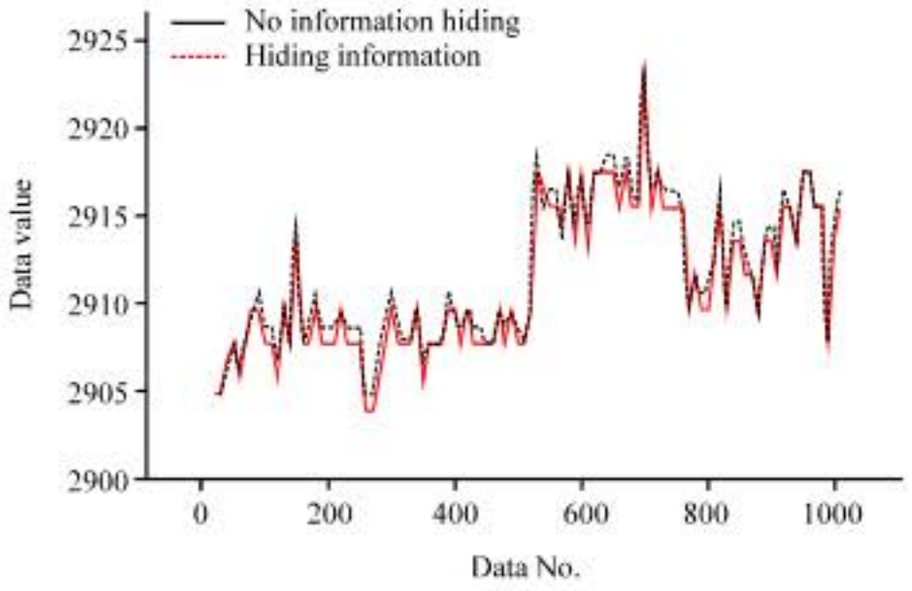

Fig. 5: Data value comparison

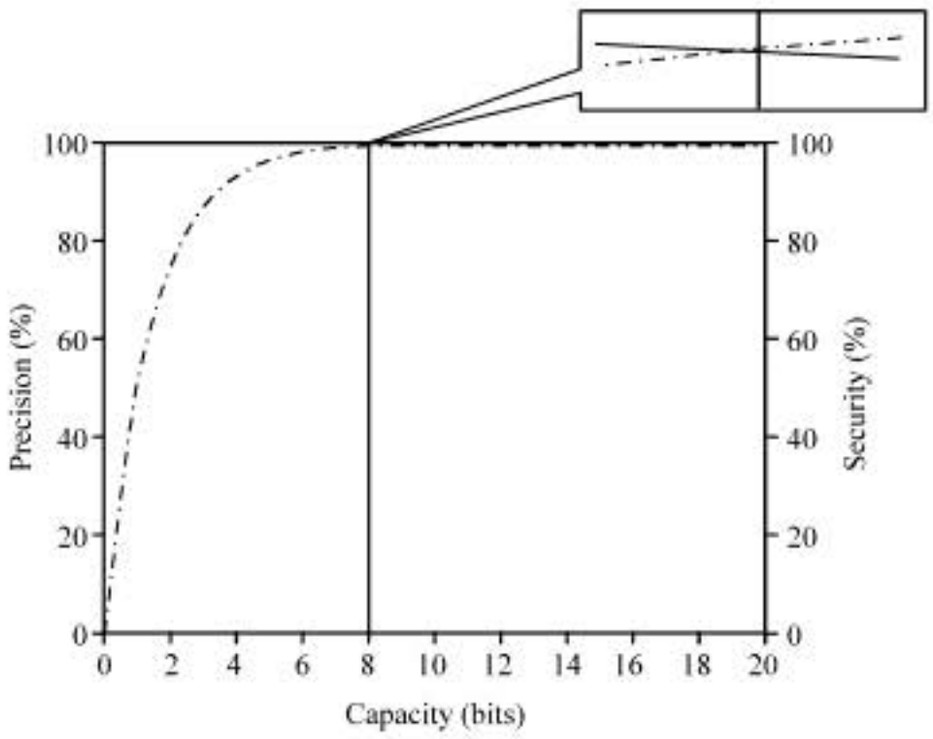

Fig. 6: Data precision and security

Let every bit b_j ($j = 0,\ldots,15$) in a datum d_i ($i = 0,\ldots,9$) be a uniform distribution, so its average is $\bar{d} = 2^{15}$. The probability of every embedded-bit being altered or not is 1/2, hence the error imported by hiding is:

$$\Delta_j = \frac{\left|d_j' - d_j\right|}{2} = 2^{n-2} \quad (1)$$

where, n is the number of embedded bits. Average data precision ($\bar{P}$) could be obtained.

$$\bar{P} = 1 - \frac{\Delta_j}{\bar{d}} = 1 - \frac{2^{n-2}}{2^{15}} = 1 - 2^{n-17} \quad (2)$$

Security measures the strength of hidden data and denotes it as S. Equation 3 shows the probability of hidden data successfully extracted by brute force.

$$S = 1 - \frac{1}{2^n} = 1 - 2^{-n} \quad (3)$$

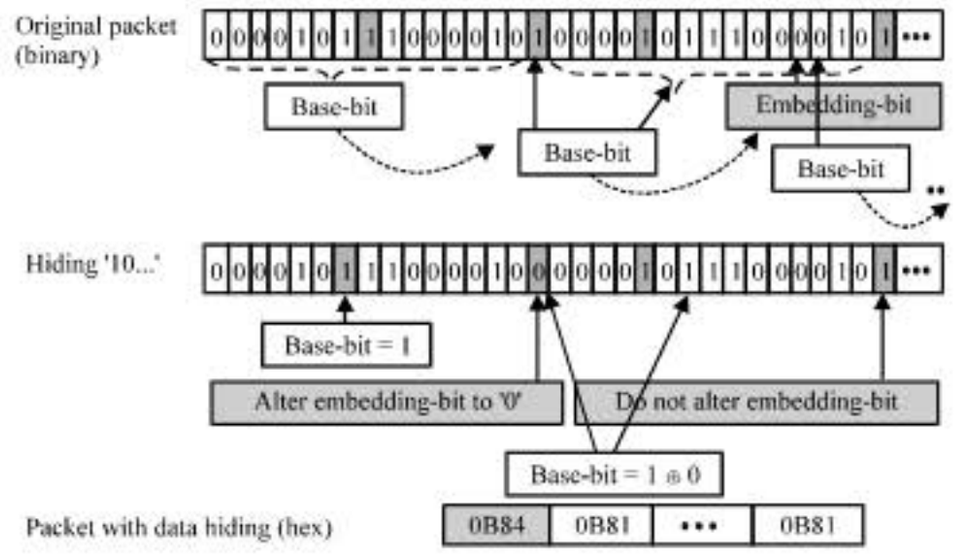

Fig. 7: Data hiding with fragile algorithm

Intuitively, it is obvious that the more capacity for embedding, the more security of our hiding scheme has, but the more precision of data is lost. Observing from Fig. 6, we obtain S = 98.44% when n = 6, S = 99.61% when n = 8 and S = 99.90% when n = 10. Therefore, we would reach better capacity and security when hiding eight bits in one packet.

Improvement: We then propose two improvements on the basic design: fragile algorithm and compact algorithm. Fragile algorithm has stronger tamper detection and compact algorithm has higher embedding capabilities.

Fragile algorithm: We convert the basic design into fragile algorithm, which differs in base-bit selection. As mentioned earlier, base-bit is the embedding reference in the first 15 bit of a datum. Now we embed the first bit to the former rule and the remaining bit is embedded according to the result of the LSB of the previous data XOR to its base-bit one by one, i.e., $b_i = b_i \oplus b_{i-1}$.

Figure 7 shows the fragile algorithm hiding implementation, while the original packet is the same as Fig. 4. The hiding marks correlate to each other in the packet, so it is more fragile than the basic algorithm. Hence, tampering can be detected even if there is little distortion.

Compact algorithm: In large scale WSNs, The node ID will be longer, leading to the necessity of hiding more marks. Therefore, we design the compact algorithm, as shown in Fig. 8.

What differs from the basic algorithm is the selection of base-bit and embedded-bit. Here we use the last n bits for embedding and the remaining 16-n bits for choosing a reference in each datum. The hiding capacity is enlarged to n times than the basic one, but at the same time, the precision is reduced.

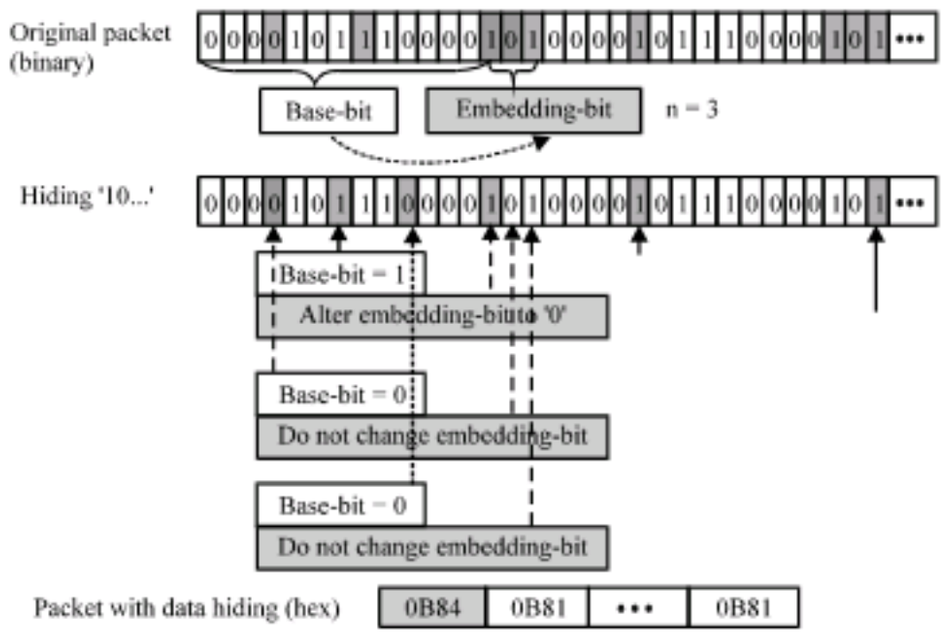

Fig. 8: Data hiding with compact algorithm

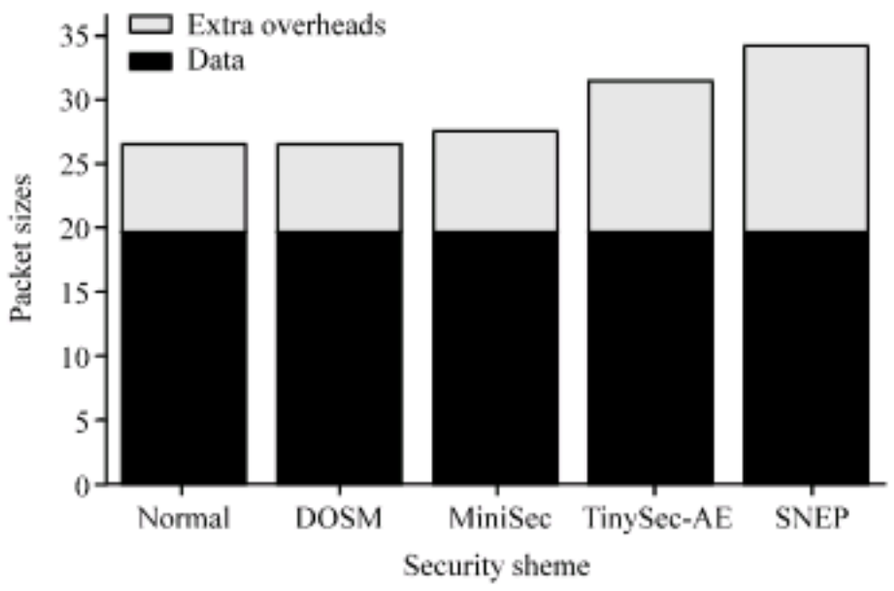

Fig. 9: Throughput comparisons

Throughput analysis: There are 7 bytes of additional overheads in the aforementioned packet. Figure 9 shows the throughput comparison with existing secure approaches (Karlof *et al*., 2004; Luk *et al*., 2007; Perrig *et al*., 2002).

Compared with other schemes, DOSM has the same as normal throughput but obtains a more secure performance. The authenticated marks do not occupy extra overheads in packets while ensuring security by IH techniques.

Invisibility analysis: The comparisons of statistical characteristics before and after hiding a mark are shown in Table 1.

Intuitively, the variations in data that mark embedding by switching the LSB of every datum cannot affect the application of acquired data. Moreover, the position of reference base-bit is computed based on the value of acquiring data, the count of packet and the secret key K. The value of sensing data and packet count are fluctuant and the key is secret, which would lead to changeable hiding positions.

Table 1:The statistical indicators comparison

Statistical value	Before hiding	After hiding
Average value	2933.13	2933.06
SD	11.056	11.067
Median	2996	2996

In addition, Figure 5 shows there is less distortion after mark embedding. Therefore, it has such a good invisibility that attackers cannot detect and localize the hidden information through statistical analysis.

Moreover, the sensing data and hidden mark are real-time, while the identification is embedded with the concealable packet. They are highly dynamic, invisible and inattentive, which decreases the possibility of attacks. Therefore, the security performance is good.

Computational complexity: It is known that most overheads arise from the transmission of extra data rather than from any computational costs (Perrig *et al*., 2002). Existing encryption techniques have higher computational complexity and longer cryptography, which is not adequate for WSNs with limited computational resources. It should be noted that most cryptography is based on the RSA algorithm, which is of polynomial time with large prime number computation.

In DES encryption, 16 rounds of encryption and a space of 64 bits are needed to cipher a text. RSA encryption is based on the modular exponentiation computation of large numbers -large numbers are usually 512-1024 bits. In this approach, the space of the hidden code is less than the data in packets and DOSM only makes use of the normal XOR computation with mark hiding and decoding/verification. Hence, the computational complexity is much lower than encryptions. Specifically, we have O(n) computation complexity in DOSM, where n is the length of hidden identification. Since in practice n is not too large, DOSM is of computational efficiency.

SIMULATIONS

Performance metrics: A well-designed security mechanism can efficiently prevent sensor nodes from various attacks. Consequently, we present some metrics to evaluate the effects of the proposed DOSM model as follows:

Accurate Received Data (ARD): Accurate Received Data (ARD) is the amount of accurate messages received by the sink. The higher the ARD, the better the performance will be.

There are some other metrics, Acquired Data (AD) and Received Data (RD). AD is the amount of data

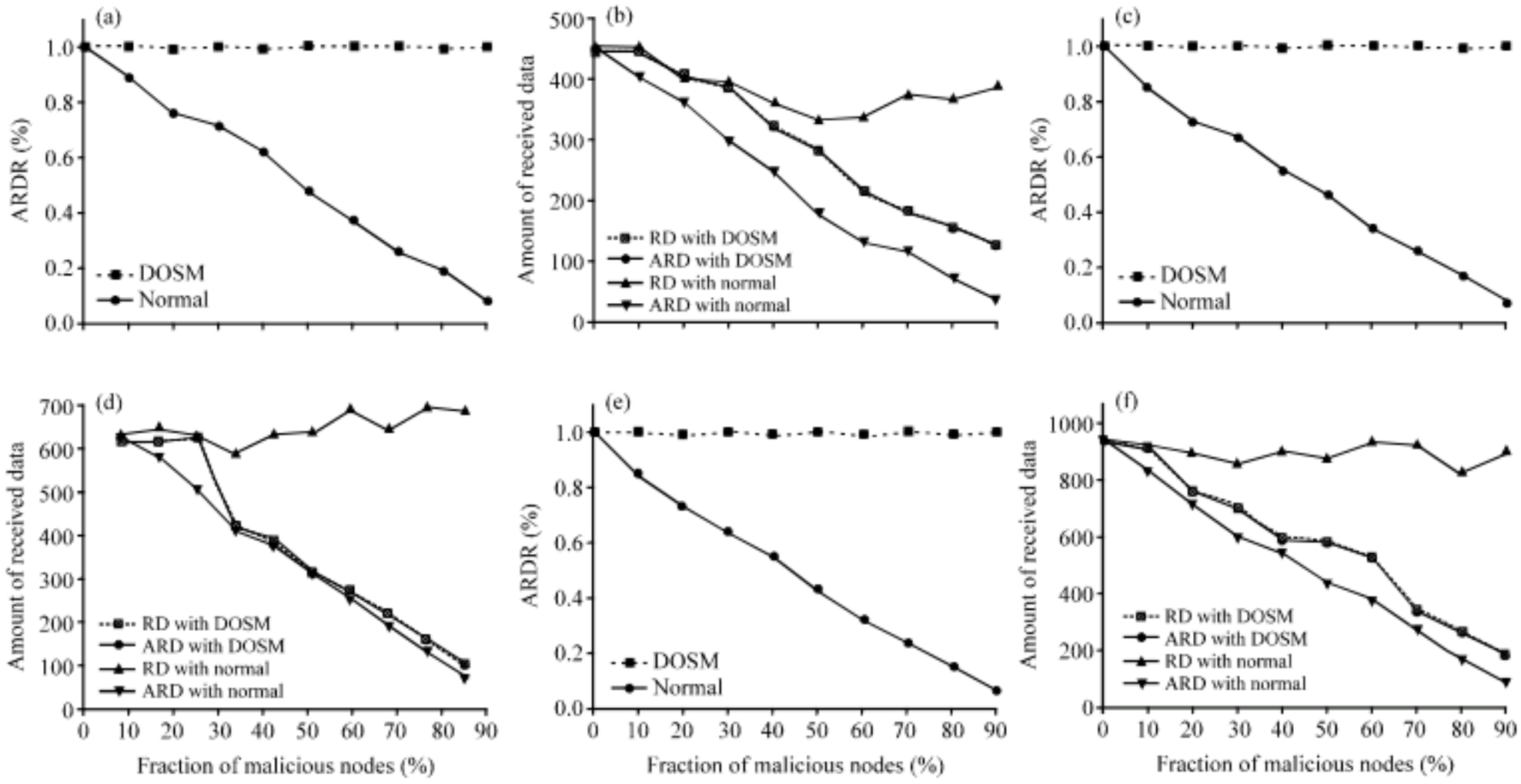

Fig. 10: Received data in the sink, (a) ARDR with 20 sensors, (b) ARD and RD with 20 sensors, (c) ARDR with 50 sensors, (d) ARD and RD with 50 sensors, (e) ARDR with 100 sensors and (f) ARD and RD with 100 sensors

generated by the source node. RD shows the amount of packets received by the sink. RD-ADR is the mount of error in received data.

Accurate Received Data Rate (ARDR): Accurate Received Data Rate (ARDR) is the fraction of accurate data received by the sink, which is computed as:

$$\text{ARDR} = \frac{\text{ARD}}{\text{RD}} \times 100\% \quad (4)$$

The higher the obtained ARDR, the better the security mechanism works.

Life time (LI): We define the lifetime as the time for the first node to fail, which equals the minimum lifetime of all nodes (Chang and Tassiulas, 2000). We think it good when LI is long, as it evaluates the efficiency combined with ARD.

Latency (LA): LA is the total delay for packets transmission and the less the better.

Simulation setup: We simulate sensor nodes with random uniform distribution in a 500×500 m area assuming that there are no error codes and dropping data. We record the behavior of sensors under both normal and attack situations. In a normal situation, there are no embedding-bit attackers; in an attack situation, the malicious nodes attempt or fake the relaying packets to disturb the communication.

Table 2: Simulation parameters

Parameters	Values
Sense area	500×500
Node numbers	20/50/100
Sense interval	10 sec
Malicious	10-90%
Initial energy	5 J
Sensing power	1.75 mW
TX/RX power	1.6/1.2 mW
Idle power	1 mW
Embedding (Pottie *et al*., 2000)	1.75 uW
Verification 1	1.75 uW

Pottie and Kaiser (2000) proposed that the energy consumption when transmitting 1 bit of data on the wireless channel is similar to thousands of cycles of CPU instructions. Hence, we set the energies of embedding and verifying to 10-3 of transmission power

The parameters of experiment are shown in Table 2, which are used for each attack.

We simulated these attacks while adjusting the fraction of malicious nodes in WSNs from 10 to 90% with different scales of WSNs (20, 50 and 100).

Performance: To evaluate the efficiency what DOSM can reach, we gather the metrics including ARD, ARDR, LI and LA in simulations.

Reliability of received data: Figure 10 shows RD, ARD and ARDR when different fractions of malicious nodes exist in different scales of WSNs.

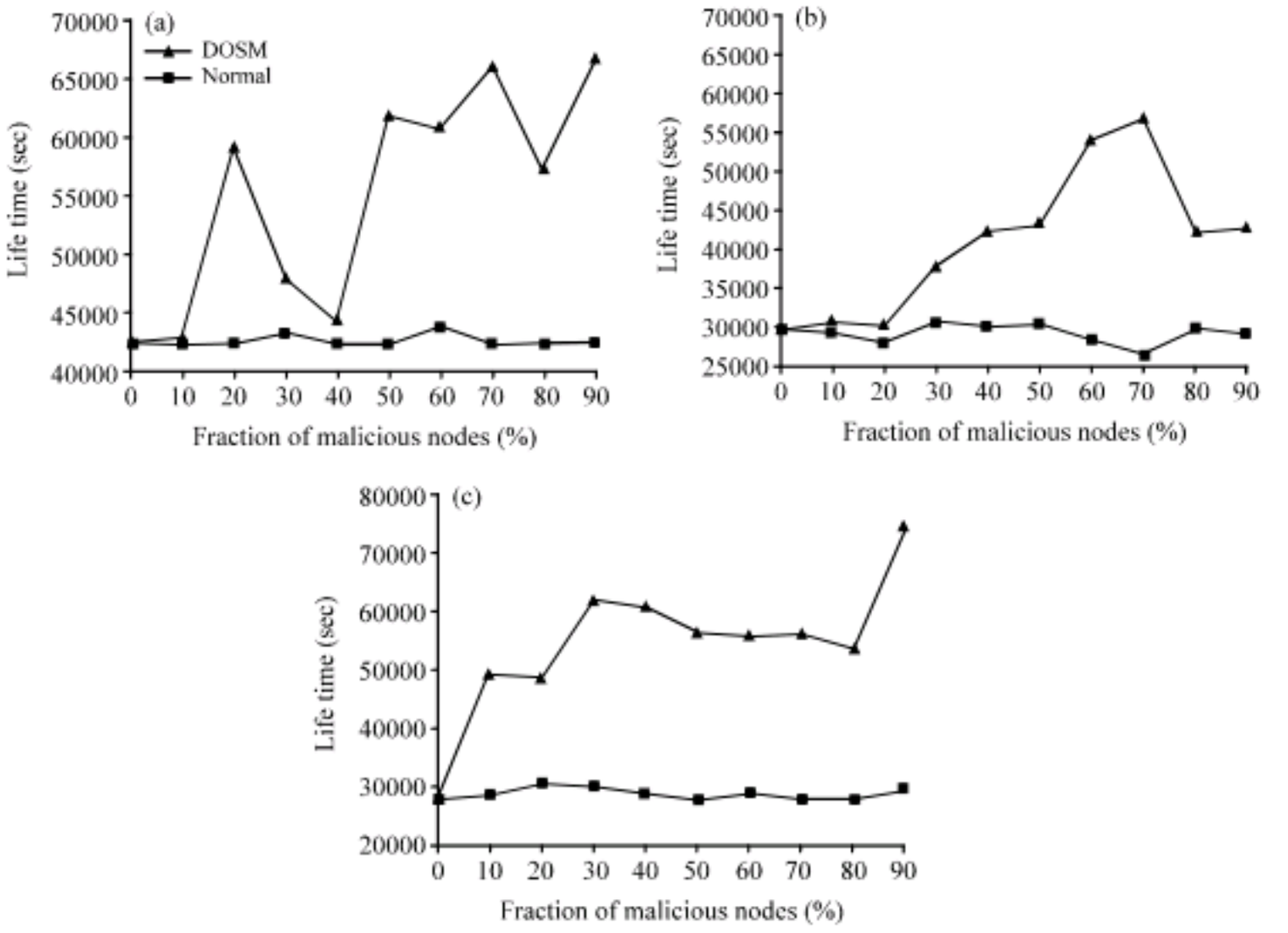

Fig. 11: Life time of WSNs, (a) lifetime with 20 sensors, (b) lifetime with 50 sensors and (c) lifetime with 100 sensors

There are 3 scales of WSNs in this simulation: 20, 50 and 100 nodes. We obtain almost 100% of ARDR using DOSM, while ARDR decreases linearly with the increase of malicious nodes. At the same time, we obtain a higher ARD adopting DOSM, which indicates that the sink can receive more accurate data.

Although, RD is higher in normal situations, most of them are useless. In summary, most received data are accurate by DOSM, i.e., ARD˜RD, while only the part of RD is accurate in normal situations, i.e., ARD<RD.

The results of attacks prove that the relay nodes filter the fake or tampered data effectively before forwarding it, which improves the security of communication performance.

Life time: A simulation has been performed to illustrate the difference in the lifetime in Fig .11.

The dashed line denotes the lifetime for applying DOSM and the solid one denotes normal. It is obvious that the lifetime is increased with the bad packet filtering using DOSM.

The lifetime is similar when there are no malicious nodes. It shows the extra energy consumption is low for embed ding and decoding/verifying, i.e., we obtain higher ARD with trivial energy consumption.

Latency: Figure 12 draws a comparison of the latency in packet transmitting. The dashed line denotes the latency adopting DOSM and the solid one with no DOSM. It is obvious that the transmitting latency increases with the scales of WSNs.

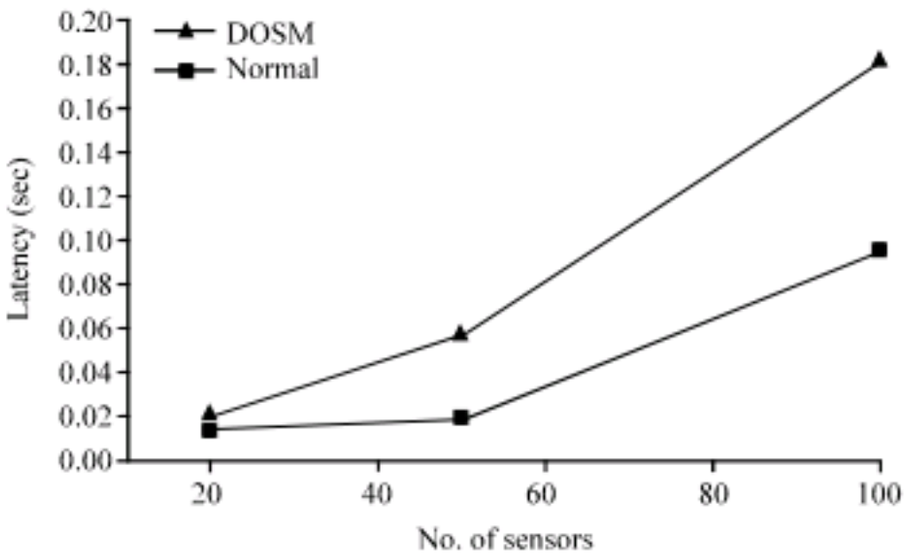

Fig. 12: Transmitting latency

The total latency contains two parts: one is the time cost in ID embedding and the other is for consistency verification before forwarding. Altogether, the computation complexity for embedding and decoding/verifying is too low to increase transmitting delay.

Apparently, DOSM bears less latency for protecting data security. The average latency is under 0.2 sec.

CONCLUSION AND FUTURE WORK

In this study, we have proposed a non-cryptology and protocol-independent model, IH technique based DOSM. With the help of DOSM, each intermediate node can verify the embedded marks with the source node ID

in the packets. The node, hence, can terminate bad packets as soon as they detect inconsistency.

DOSM enhances the security of a sensor network with higher reliability, better invisibility and lower computation complexity. It is robust to a certain set of attacks including packet forging and eavesdropping. Also, it is transparent to all applications and can be applied on various motes with different microprocessors and radios.

We aim to do research on security for physical capture defending and take powerful action against malicious nodes in the future. More mechanisms need to be applied to the security aspects of WSNs.

ACKNOWLEDGMENTS

This study is supported by National Basic Research Program of China (973 Program, No. 2006CB303000), National High Technology Research and Development Program of China (863 Program, No. 2007AA01Z180), National Natural Science Foundation of China (No. 60573045 and 60873198), NSFC Key Project Grant (No. 60736016).

REFERENCES

Akyildiz, I.F., W. Su, Y. Sankarasubramaniam and E. Cayirci, 2002. A survey on sensor networks. IEEE Commun. Maga., 40: 102-114.

Bremler-Barr, A. and H. Levy, 2005. Spoofing prevention method. Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Mar. 13-17, IEEE Xplore, London, pp: 536-547.

Buonadonna, P., J. Hill and D. Culler, 2001. Active message communication for tiny networked sensors. http://www.tinyos.net/papers/ammote.pdf.

Chang, J.H. and L. Tassiulas, 2000. Energy conserving routing in wireless ad hoc networks. Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies, 2000, IEEE Computer Society Press, pp: 22-31.

Deb, B., S. Bhatnagar and B. Nath, 2003. Information assurance in sensor networks. Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications, 2003, San Diego, CA, USA., pp: 160-168.

Feng, J. and M. Potkonjak, 2003. Real-time watermarking techniques for sensor networks. Proc. SPIE., 5020: 391-402.

Ford, B., 2007. Structured streams: A new transport abstraction. SIGCOMM, 37: 361-372.

Guo, H.P., Y.J. Li and S. Jajodia, 2007. Chaining watermarks for detecting malicious modifications to streaming data. Inform. Sci., 177: 281-298.

Karlof, C., N. Sastry and D. Wagner, 2004. TinySec: A link layer security architecture for wireless sensor networks. Proceedings of the 2nd international conference on Embedded networked sensor systems, Nov. 03-05, Baltimore, MD, USA., pp: 162-175.

Kleider, J.E., S. Gifford, S. Chuprun and B. Fette, 2004. Radio frequency watermarking for OFDM wireless networks. Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 17-21, IEEE Xplore, London, pp: 397-400.

Liu, A. and P. Ning, 2008. TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks. Proceedings of the 7th International Conference on Information Processing in Sensor Networks, Apr. 22-24, IEEE Xplore, London, pp: 245-256.

Luk, M., G. Mezzour, A. Perrig and V. Gligor, 2007. MiniSec: A secure sensor network communication architecture. Proceedings of the 6th International Conference on Information Processing in Sensor Networks, Apr. 25-27, Cambridge, Massachusetts, USA., pp: 5479-488.

Mitrea, M., S. Duta and F. Prêteux, 2008. Proving video stream watermarking viability. SPIE Newsroom. 10.1117/2.1200806.1174

Ning, P., A. Liu and W.L. Du, 2008. Mitigating DoS attacks against broadcast authentication in wireless sensor networks. ACM Trans. Sensor Networks (TOSN), 4: 1-35.

Perrig, A., R. Szewczyk, J.D. Tygar, V. Wen and D.E. Culler, 2002. SPINS: Security protocols for sensor netowrks. Wireless Networks, 8: 521-534.

Petitcolas, F.A.P., R.J. Anderson and M.G. Kuhn, 1999. Information hiding-a survey. Proc. IEEE, 87: 1062-1078.

Pottie, G.J. and W.J. Kaiser, 2000. Embedding the internet: Wireless integrated network sensors. Commun. ACM, 43: 51-58.

Sion, R., M. Atallah and S. Prabhakar, 2006. Rights protection for discrete numeric streams. IEEE Trans. Knowledge Data Eng., 5: 1-16.

Smith, J.R., B. Jiang, R. Sumit and P. Matthai, 2005. ID modulation: Embedding sensor data in an RFID timeseries. Lecture Notes Comput. Sci., 3727: 234-246.