SecMAS: Security Enhanced Monitoring and Analysis Systems for Wireless Sensor Networks

Chao DING[1], Li-Jun YANG[2] and Meng WU[3,a]

[1]*College of Computer, Nanjing University of Posts and Telecommunications, 210003 Nanjing, China*
[2]*College of Internet of Things, Nanjing University of Posts and Telecommunications, 210003 Nanjing, China.*
[3]*College of Telecommunications & Information Engineering, Nanjing University of Posts and Telecommunications, 210003 Nanjing, China*

Abstract. The monitoring, control, and security guarantee for the communication in the wireless sensor networks (WSNs) are currently treated as three independent issues and addressed separately through specialized tools. However, most cases of WSNs applications requires the network administrator change the network configuration in a very short time to response to the change of observed phenomenon with security guarantee. To meet this requirement, we propose a security enhanced monitoring and control platform named SecMAS for WSNs, which provides the real-time visualization about network states and online reconfiguration of the network properties and behaviours in a resource-efficient way. Besides, basic cryptographic primitives and part of the anomaly detection functionalities are implemented in SecMAS to enabling the secure communication in WSNs. Furthermore, we conduct experiments to evaluate the performance of SecMAS in terms of the latency, throughput, communication overhead, and the security capacity. The experimental results demonstrate that the SecMAS system achieves stable, efficient and secure data collection with lightweight quick-response network control.

1 Introduction

Compared with traditional wired and wireless networks, low-power wireless sensor networks (WSNs) can be rapidly deployed in a large geographical area in a self-configured manner. This makes them particular suitable for real-time, large-scale information collection and event monitoring for mission-critical application in hostile environment. In most scenarios, WSNs are thought to be highly coupled with the physical environment. For instance, an event occurred in the observed region may cause dramatic changes of the traffic pattern, and the network should immediately reconfigure its parameters and control its sensors' behaviours to adapt these changes.

To meet this requirement, other than tracking the network state information such as network health and diagnosis data which are usually involved in the conventional network monitoring applications, the WSNs application needs to collect the detailed runtime state data of each sensor node in the network, and reconfigure the node behaviour strategies according to alteration of the physical environments. Furthermore, the network state information is usually security-sensitive in mission-critical applications, security guarantee should be included in WSNs data collection applications. Hence, a novel data collection and analysis framework which integrates the secure communication, real-time state query and network behaviour manipulation is essential for WSNs.

There are three main challenges existing in the research and development of WSNs monitoring: (1) real-time state tracing becomes more difficult when the network scale becomes larger due to the limited on-board resource of sensor node. (2) the dependence on the air reprogram widely existing in the current WSNs network management technologies constrain the efficiency and flexibility of the network reconfiguration. (3) the lack of security functionalities makes the sensory data and network configuration information operated without any secure guarantee in most WSNs monitoring applications.

To address these challenges, a variety of research efforts are made in the field of WSNs monitoring, most of which focus on data collection and remote code dissemination. Philips et. al propose the first data collection solution Surge [1] which works on a typical hierarchical network topology and support various types of sensory data (temperature, humidity, etc.). But this solution only support the TinyOS based Mint [2] route protocol. Mviz [3] is then developed based on Surge to strengthen the protocol compatibility. On

[a] Corresponding author: wum@njupt.edu.cn

the other hand, many schemes are proposed to enhance the performance of code dissemination such as the multihop over-the-air programing (MOAP) [4] and its improved version multihop network reprogramming (MNP) [5]. Besides, another state-of-art Deluge [2] is proposed to enable administrator to update the runtime code of remote sensor nodes. In this scheme each node which receives runtime code image broadcasts advertisement in the neighbourhood and on demand forwards the code image to the neighbour nodes. But the scheme is vulnerable to the malicious code injection attack. SLUICE [6] is then proposed to add the security guarantee Deluge. The code dissemination techniques enable management systems to flexibly update the remote sensor node configuration. However this type of techniques brings frequently code image transmission over the entire network, leading to high communication overhead.

To address the challenges and the limitation of existing WSNs monitoring schemes, we propose a security enhanced monitoring and control platform named SecMAS for WSNs, which provides the real-time visualization about network states and online reconfiguration of the network properties and behaviours in a resource-efficient way. Besides, basic cryptographic primitives and part of the anomaly detection functionalities are implemented in SecMAS to enabling the secure communication in WSNs.

2 System Design

2.1 Design Overview

To adapt the tight coupling of WSNs with the deployed environment, the proposed scheme is designed to satisfy the following system requirements:

Portability and scalability: The functionality of data visualization and network configuration of SecMAS is designed to work independent of the lower-layer communication protocols such as CTP [7] and ZigBee .

Hybrid network configuration: we adopt the centralized configuration strategy in basestation-end to ensure the control accuracy from a global perceptive. Meanwhile we adopt decentralized control strategy in sensornode-end to fasten the response to the change of observed target.

Integration of cryptographic and cryptography-free security guarantee: The proposed scheme introduce lightweight public key cryptographic technologies to mitigate the computation burden on the sensor nodes. Whereas it also adopt anomaly detection based techniques to resist the inside attacks launched by compromised nodes.

2.2 Software Architecture of SecMAS System

In the overall software design phase, we divide the functionalities of the proposed SecMAS system into three different components: RemoteMote, GatewayMote and Server, as illustrated in Figure 1.

The RemoteMote module takes the responsibility of the data collection, signalling transmission and local node configuration on the sensor nodes. In SecMAS, we provide two different versions of WSNs protocol (e.g. Trickle [1], Zigbee , etc.) implementation based on TI Z-stack and TinyOS [8] libraries.

The GatewayMote module takes responsibility of the protocol transform between network communication protocols and serial communication protocols, as well as the additional services such as data cache and resource allocation. In this work, we implement a message format generator (MFG) and network parameter generator (NPG) to shield the low-layer implementation difference, and generate a unified packet and parameter format.

The Server module takes responsibility of the data intelligent analysis, geographic visualization, historical data store and GUI interface. In this work, we implement the Server module on Java 2 standard edition (J2SE) and handle the data interaction between Server and local serial port using Java native interface (JNI) technique.

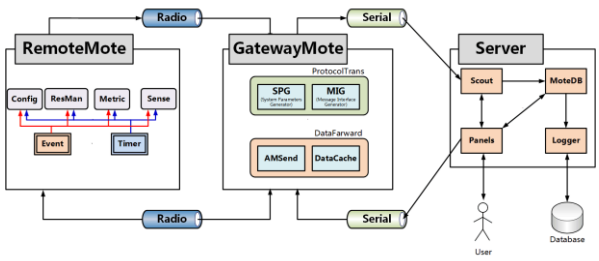
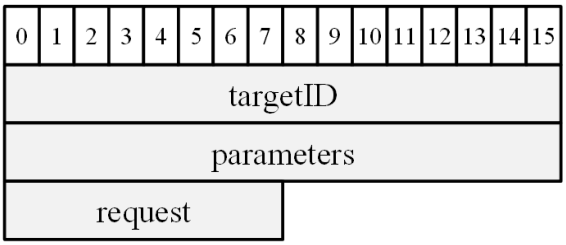


Figure 1. Software Architecture of the Proposed SecMAS

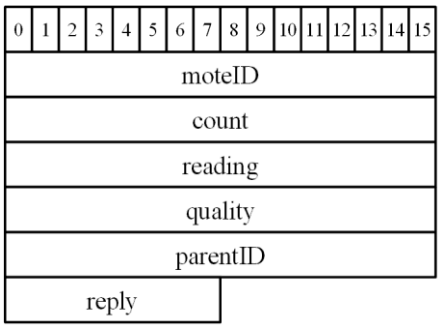
2.3 Definition of Unified Message Format

Unlike the existing WSNs monitoring schemes such as Surge and Mviz, the proposed SecMAS adopt a specific application layer message structure independent of lower-layer communication protocols. In the default case, SecMAS adopts Zigbee for information collection and Trickle for signalling dissemination. Accordingly SecMAS provides two categories of message formats: signalling message and data message format.

As illustrated in Figure 2(a), the signalling message includes a 16-bit targetID field which accounts for target node ID, a 8-bit request field which is used for specifying the signalling type, and 16-bit parameter field which stores the parameter that BS node intends to notify the target nodes. Note that regarding the length of request field, SecMAS support up to 256 types of signalling. For now the SecMAS has 15 types of signalling.



(a) Signalling Message Format



(b) Data Message Format

Figure 2. Definition of Unified Message Format

As shown in Figure 2(b), the data message includes five 16-bit fields, namely moteID, count, reading, quality, parentID and one 8-bit field reply. The field moteID represents the ID of the node which collects the data whereas the field parentID is active to represent the parent node of the source node in a hierarchical topology. The field quality account for the link quality between the source and parent nodes. The field reply represents the type of data message. In SecMAS, there are three different types of data message: reading, state info and signalling response. The field reading represent the data payload of the data message. Note that unlike other fields, parentID and quality are lower-layer protocol dependent since the quality field requires that the link layer protocol provides the link quality evaluation approach while the parentID field requires that the route protocol support the inquiry of parentID.

3 Prototype Implementation

3.1 RemoteMote Module Implementation

RemoteMote module which works on the ordinary sensor nodes, takes charge of the routine tasks of sensor nodes. A typical RemoteMote module is constructed of three different functionality components Timer, Event and Core, where Timer component calibrates the nodes' local clock, Event component handle both the hardware and software interruption requests. With the help of Timer and Event, RemoteMote is able to manipulate the rate of data collection and transmission.

In contrast, the Core component is much more complex. Its functionalities can be further divided into three subcomponents: Sense, ResMan and Config. The Sense subcomponent defines the interface of data collections, specifies the data collection mode, and generates the final data collection results. The ResMan subcomponent leverages the node into some sleep level or wake up the node on demand according to the user's specification, and report the remaining energy to BS. The Config subcomponent update local configuration and adjust the sensor node's behaviours based on the signalling message from BS.

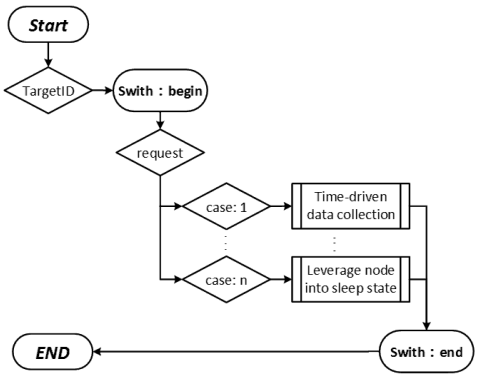


Figure 3. Workflow of the Processrequest Function

Additionally, Core component has a multiple selection function processRequest so as to invoke related functionality components such as Core, ResMan, and Config. The workflow of the function processRequest is illustrated in Figure 3.

3.2 Gateway Module Implementation

Gateway module which plays the role of the bridge between RemoteMote and Server modules, takes responsibility of reception the data from RemoteMote module, encapsulation of the received data, and transmission of the encapsulated objects.

Gateway module is constructed of two functionality components: DataForward and ProtocolTrans, where the DataForward component takes charge of data forwarding, caching, and data rate adjustment between different protocols. Whereas the ProtocolTrans component takes charge of protocol transformation.

3.3 Server Module Implementation

Server module which works on the BS node in the sensor networks, take responsibilities of the recording and storage of the readings and node state information, signalling dissemination and real-time data visualization. In this work, we implement the Server module using J2SE following the classic Model-View-Controller (MVC) design pattern, where *Model* handles the storage of events, readings and state information in the RAM and NAND Memory. *Controller* takes charge of data interaction including the

interaction with Gateway module, interaction with database, interaction with graphic user interface (GUI). *View* represents the user request and feedback from GUI, and the dynamic data visualization.

Based on the adopted MVC design pattern, the functionalities of Server module is divided into several related classes with least coupling, as shown in Figure 4.

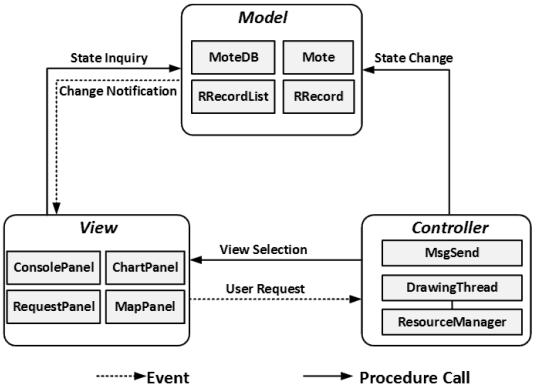


Figure 4. The MVC Design Pattern and Functionality Division Of **Server** Module.

3.4 Implementation of Security Guarantee

3.4.1 *Cryptography-Based Security Guarantee*

To achieve cryptographic security guarantee for WSNs, we develop a resource efficient cryptography library named LWCrypt based on Lynn's Pair Based Cryptography (PBC) library for SecMAS. We rewrite part of application program interface (API) in Lynn's library to eliminate the dependency with the Linux standard library such as GMP and OpenSSH, enabling the usage of LWCrypt on resource-constrained sensor network hardware.

We found that the basic cryptography operations large integer modular (LIMR), large integer multiplication (LIMS), elliptic curve scalar multiplication (ECSM) and bilinear pairing (BP) consume more than 70% computation resource in more than 90% cases during the runtime of LWCrypt. Thus it is important to reduce the computation complexity of LIMR, LIMS, ECSM, and BP.

To minimize the computation overhead of LIMR in the prime field, we adopt Berrett Reduction [9] algorithm instead of basic division operation to transform the LIMR to twice LIMS and 2^n modular operation which is much more lightweight in prime field.

To reduce the complexity of LIMS, Instead of storing the base and order in the RAM which is common in the conventional protocols like IEEE754, we adopt the Hybrid Multiplication algorithm [10] to enhance the efficiency by optimizing the usage of microprocessor registers, which significantly reduce the frequency of the interoperation between registers and RAM, leading to accelerate the cryptographic operations.

Since scalar multiplication on elliptic curves in the affine coordinate system requires resource-consuming *modular inverse* operation, we transfer modular inverse to several resource-efficient *modular multiplication* in the projection coordination system, and further adopt the Mix Point Addition and Repeated Doubling algorithms to enhance the performance of ECSM.

For BP, we choose appropriate elliptic curves to achieve best computation speed and memory allocation. In this work, we select the super singular elliptic curve $y^2 + y = x + x^3$ in the $\mathbf{F}_{2^{273}}$ binary field.

3.4.2 *Anomaly Detection Based Security Guarantee*

To defender the attacks from inside compromised nodes, we develop an anomaly detection based security algorithm library (ADSAL) for SecMAS. As illustrated in Figure 5, the architecture of ADSAL is constructed of four functionality components: DataFeatureExtractor, LocalDetector, ReportHandler and GlobalDetector.

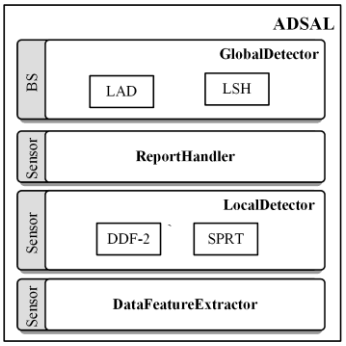


Figure 5. The **MVC** Design Pattern and Functionality Division Of Server Module.

The DataFeatureExtractor component provides the functionalities of data feature extraction and redundancy compression. This component parses the feature information for the further analysis of LocalDetector and GlobalDetector. The LocalDetector provides the functionalities of decentralized anomaly detection running on the local sensor nodes. In current version of SecMAS, we have implemented the the second-order divided difference filtering (DDF-2) [11] state estimation algorithm, the sequential probability ratio testing (SPRT) [12] based decision strategy. The GlobalDetector provides the functionalities of centralized anomaly detection. The implemented components in current version include the deployment knowledge based local anomaly detection (LAD) algorithm and locality sensitive hash (LSH) [13] algorithm.

4 Performance Evaluation

4.1 Experiment Setup and Methodology

We evaluate the performance of the proposed scheme in terms of data collection and network configuration. In order to establish a experimental network, we deploy 5 telosb motes of which one works as BS, and 28 MicaZ motes in a $10 \times 6\ m^2$ area. Since SecMAS is able to remotely force the active nodes into sleep and wake up sleeping nodes, we can control the network scale by limit the number of active nodes.

We conduct two group of experiments, the first group evaluates impact of the node sampling frequency on the packet delivery rate, the second group evaluates the impact of duty cycle on the packet delivery rate.

4.2 Results and Discussion

We firstly study the varying sampling period and network scale on the packet delivery rate while fixing the duty cycle at 100%. We present the tendency of delivery rate while the sampling period varying from 1 second to 11 seconds whereas the number of sensor nodes varying from 10 to 32 in Figure 6. We notice that the delivery rate becomes higher when the sampling period becomes lower, which indicates that the congestion and collision happen much more easily when the behaviour of sampling is performed more frequently.

When the number of sensor nodes is fixed at 10, the delivery rate always keeps at approximately 100% while the sampling period varying from 1 to 11. However, the delivery rate is much lower while the sampling period varying from 1 to 6 when the number of sensor nodes is 32 compared with that when the number of sensor nodes is 10. This is because that channel contention in the neighbourhood becomes more frequently when network scale becomes larger.

Then we study the impact of duty cycle on the delivery rate while fixing the number of sensor nodes at 32. We present the tendency of delivery rate while the duty cycle varying from 1% to 100%, in Figure 7. We notice that the delivery rate increases with the rise of duty cycle. That is because when the value of duty cycle becomes higher, the sleep time becomes shorter, the sensor node has more efficient time to receive and forwards packets, leading the decrease of probability that the packets are dropped during the transmission. For example, the delivery rate is only 77% when the cycle duty is 1%, that is because the transceiver of each sensor node is at sleep in the 99% cases.

When the value of duty cycle ranges from 10% to 20%, the delivery rate increase linearly, and comes to 78.5% and 80.7%. However when the value of duty cycle further increases and exceeds 50%, the growth rate of delivery rate dramatically decreases. When the value of duty cycle ranges from 60%-100%, the delivery rate reaches a local peak at 70% with high variance among the experiment results. This means that the transceiver receives large amount of forwarding requests from the neighbour nodes, which causes severe congestion, leading to an increase of the data delivery rate. Furthermore, we infer that the high variance of packet delivery rate at certain values of duty cycle comes from some sensitivity of MAC and routing mechanisms in the protocols adopted by SecMAS.

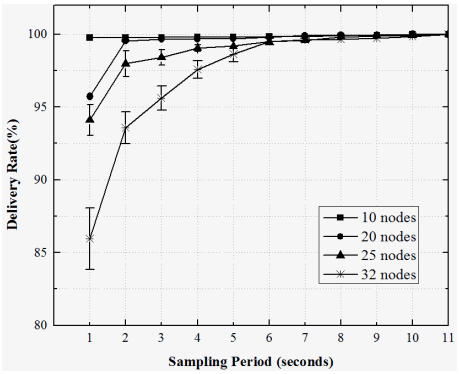


Figure 6. Impact of Sampling Period and Network Scale on The Packet Delivery Rate.

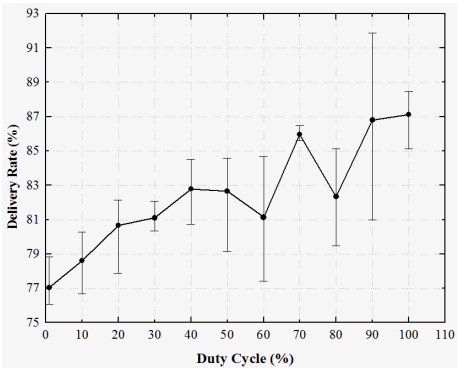


Figure 7. Impact of Duty Cycle On The Packet Delivery Rate.

5 Conclusions

In this work, we propose a security enhanced monitoring and analysis system SecMAS for WSNs to address the limitations including real-time monitoring and manipulation of the single node, inflexibility of network reconfiguration, and absence of security guarantees in the existing schemes. We present the design goals and software architecture, and further describe the implementation details of the key functionality components. Additionally, we evaluate the performance of the propose scheme through the experiments. The results demonstrate that SecMAS provide flexible strategies of network reconfiguration, and achieves promising network connectivity through appropriate network configuration.

Acknowledgment

This work is supported by National Basic Research Program of China (973 Program) under Grants 2011CB302903, the Natural Science Foundation of Jiangsu Province (Grant NO. BK20151507), the Natural Science Foundation of Jiangsu Province for Youth (Grant No. BK20160916), , the Key Program of Natural Science for Universities of Jiangsu Province (Grant No.10KJA510035).

References

1. L. Philips, Patel N., and Culer D., Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks, Proc. NSDI'04, (2004).
2. J. Hui. TinyOS Network Programming [Online]. Available: http://www.cs.berkeley.edu/jwhui/deluge/
3. Mviz [Online]. Available: http://www.tinyos.net/tinyos-2.x/apps/Mviz
4. T. Stathopoulos, T. McHenry, J. Heidemann, and D. Estrin, Remote code update mechanism for wireless sensor networks, CENS Technical Report #30, (2003).
5. S. Kulkarni and L. Wang, MNP: multihop network programming for sensor networks, Proc. ICDCS'05 (2005).
6. P. Lanigan, R. Gandhi, and P. Narasimhan, Sluice: secure dissemination of code update in sensor netowrks, Proc. ICDCS'06, (2006).
7. P. Levis, N. Lee, M. Welsh, and D. Culler, TOSSIM: accurate and scalable simulation of entire TinyOS applications, Proc. Sensys'03, (2003).
8. L. Philips, M. S, and D. Gay, The emergence of networking abstractions and techniques in TinyOS, Proc. NSDI'04, (2004).
9. P. Barrett, Implementing the Rivest Shamir and Adleman Public Key Encryption Algorithm on a Standard Digital Signal Processor, CRYPTO'86, 311-323, (1986).
10. L. An and N. Peng, TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks, Proc. IPSN'08, 245-256, (2008).
11. M. Nørgaard, N. K. Poulsen, and O. Ravn, New developments in state estimation for nonlinear systems, Automatica, 1627-1638, 36(2000).
12 . A. Wald, Sequential Tests of Statistical Hypothesis," The Annals of Mathematical Statistics, 117-186, (1945).
13. M. Charikar, Similarity estimation techniques from rounding algorithms, Proc.STC, 380-388, (2010).