Energy and Memory Efficient Clone Detection in Wireless Sensor Networks

Mirza Moiz Baig[1], Umesh Samarth[2], Karishma Misal[3], Mikhal John[4], Ekta Parsawani[5]

[1,2,3,4]Assistant Professor, Department of Information Technology J D College of Engineering & Management, Nagpur

[5]Student Department of Information Technology J D College of Engineering & Management, Nagpur

ABSTRACT

In this paper, we propose an energy-efficient location-aware clone detection protocol in densely deployed WSNs, which can guarantee successful clone attack detection and maintain satisfactory network lifetime. Specifically, we exploit the location information of sensors and randomly select witnesses located in a ring area to verify the legitimacy of sensors and to report detected clone attacks. The ring structure facilitates energy-efficient data forwarding along the path towards the witnesses and the sink.We theoretically prove that the proposed protocol can achieve 100 percent clone detection probability with trustful witnesses. We further extend the work by studying the clone detection performance with untrustful witnesses and show that the clone detection probability still approaches 98 percent when 10 percent of witnesses are compromised. Moreover, in most existing clone detection protocols with random witness selection scheme, the required buffer storage of sensors is usually dependent on the node density, i.e., O(n−−√) , while in our proposed protocol, the required buffer storage of sensors is independent of n but a function of the hop length of the network radius h , i.e., O(h) . Extensive simulations demonstrate that our proposed protocol can achieve long network lifetime by effectively distributing the traffic load across the network.

KEYWORDS: Sensors, Clone, Protocol

1. INTRODUCTION

Wireless sensors have been widely deployed for a variety of applications, ranging from environment monitoring to telemedicine and objects tracking, etc. For cost-effective sensor placement, sensors are usually not tamper-proof devices and are deployed in places without monitoring and protection, which makes them prone to different attacks. For example, a malicious user may compromise some sensors and acquire their private information.

Then, it can duplicate the sensors and deploy clones in a wireless sensor network (WSN) to launch a variety of attacks, which is referred to as the clone attack. As the duplicated sensors have the same information, e.g., code and cryptographic information, captured from legitimate sensors, they can easily participate in network operations and launch attacks.

Due to the low cost for sensor duplication and deployment, clone attacks have become one of the most critical security issues in WSNs. Thus, it is essential to effectively detect clone attacks in order to ensure healthy operation of WSNs. To allow efficient clone detection, usually, a set of nodes are selected, which are called witnesses, to help certify the legitimacy of the nodes in the network. The private information of the source node, i.e., identity and the location information are shared with witnesses at the stage of witness selection.

When any of the nodes in the network wants to transmit data, it first sends the request to the witnesses for legitimacy verification, and witnesses will report a detected attack if the node fails the certification. To achieve successful clone detection, witness selection and legitimacy verification should fulfill two requirements: 1) witnesses should be randomly selected; and 2) at least one of the witnesses can successfully receive all the verification message(s) for clone detection.

The first requirement is to make it difficult for malicious users eavesdrop the communication between current source node and its witnesses, so that malicious users cannot generate duplicate verification messages. The second requirement is to make sure that at least one of the witnesses can check the identity of the sensor nodes to determine whether there is a clone attack or not. To guarantee a high clone detection probability, i.e., the probability that clone attacks can be successfully detected, it is critical and challenging to fulfill these requirements in clone detection protocol design. Different from wireless terminal devices, wireless sensors are usually of smaller size and lower price, and have limited battery and memory capacity. Therefore, the design criteria of clone detection protocols for sensor networks should not only guarantee the high performance of clone detection probability but also consider the energy and memory efficiency of sensors. In the literature, some distributed clone detection protocols have been proposed, such as Randomized Efficient and Distributed protocol (RED) and Line Select Multicast protocol (LSM).

However, most approaches mainly focus on improving clone detection probability without considering efficiency and balance of energy consumption in WSNs. With such kind of approaches, some sensors may use up their batteries due to the unbalanced energy consumption, and dead sensors may cause network partition, which may further affect the normal operation of WSNs.

To prolong network lifetime, i.e., time duration from the start of network until the first occurrence of a sensor that runs out of energy, it is critical to not only minimize the energy consumption of each node but also balance the energy consumption among sensors distributive located indifferent areas of WSNs. The limited memory or data buffer is another important feature of sensors which has significant impact on the design of clone detection protocols. Generally, to guarantee successful clone detection, witnesses need to record source nodes' private information and certify the legitimacy of sensors based on the stored private information.

In most existing clone detection protocols, the required buffer storage size depends on the network node density, i.e., sensors need a large buffer to record the exchanged information among sensors in a high-density WSN, and thus the required buffer size scales with the network node density.

Protocol is applicable to general densely deployed multi-hop WSNs, where adversaries may compromise and clone sensor nodes to launch attacks. A preliminary work is presented. In that work, we proposed an energy-efficient ring based clone detection (ERCD) protocol to achieve high clone detection probability with random witness selection, while ensuring normal network operations with satisfactory network lifetime of WSNs. The ERCD protocol can be divided into two stages: witness selection and legitimacy verification. In witness selection, the source node sends its private information to a set of witnesses, which are randomly selected by the mapping function. In the legitimacy verification, verification message along the private information of the source node is transmitted to its witnesses. If any of witnesses successfully receives the message, it will forward the message to its witness header for verification. Upon receive the messages; the witness header compares the aggregated verification messages with stored records.

If multiple copies of verification messages are received, the clone attack is detected and a revocation procedure will be triggered. As such, to have a comprehensive study of the ERCD protocol, we extend the analytical model by evaluating the required data buffer offered protocol and by including experimental results to support our theoretical analysis. First, we theoretically prove that our proposed clone detection protocol can achieve probability 1 based on trustful witnesses. Considering the scenario that witnesses can be compromised, our simulation results demonstrate that the clone detection probability can still approach 98 percent in WSNs with 10 percent cloned nodes by using the ERCD protocol. Second, to evaluate the performance of network lifetime, we derive the expression of total energy consumption, and then compare our protocol with existing clone detection protocols. We find that the ERCD protocol can balance the energy consumption of sensors at different locations by distributing the witnesses allover WSNs except non-witness rings, i.e., the adjacent rings around the sink, which should not have witnesses. After that, we obtain the optimal number of non-witness rings based on the function of energy consumption.

Finally, we derive the expression of the required data buffer by using ERCD protocol, and show that our proposed protocol is scalable because the required buffer storage is dependent on the ring size only. Extensive simulation results demonstrate that our proposed ERCD protocol can achieve superior performance in terms of the clone detection probability and network lifetime with reasonable data buffer capacity.

Literature:

Randomized Multipath Routing for Secure Data Collection

WSN (Wireless Sensor Network) has a wide range of applications. As a result, security problems become increasingly important. We investigate the problem of minimizing the failure rate of packet delivery in the presence of the modification attacks and the selective forwarding attacks in a static WSN with one base station without using expensive encryption/decryption algorithms. We propose a novel heuristic approach to this problem.

Design principles and improvement of cost function based energy aware routing algorithms for wireless sensor networks

Cost function based routing has been widely studied in wireless sensor networks for energy efficiency improvement and network lifetime elongation. However, due to the complexity of the problem, existing solutions have various limitations. In this paper, they analyze the inherent factors, design principles and evaluation methods for cost function based routing algorithms. Two energy aware cost based routing algorithms named Exponential and Sine Cost Function based Route (ESCFR) and Double Cost Function based Route (DCFR) have been proposed in this paper. For ESCFR, its cost function can map small changes in nodal remaining energy to large changes in the function value.

Analysis

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system the software development process.

Existing System

Most approaches mainly focus on improving clone detection probability without considering efficiency and balance of energy consumption in WSNs. With such kind of approaches, some sensors may use up their batteries due to the unbalanced energy consumption, and dead sensors may cause network partition, which may further affect the normal operation of WSNs. To prolong network lifetime, i.e., time duration from the start of network until the first occurrence of a sensor that runs out of energy, it is critical to not only minimize the energy consumption of each node but also balance the energy consumption among sensors distributively located in different areas of WSNs.

The limited memory or data buffer is another important feature of sensors which has significant impact on the design of clone detection protocols. Generally, to guarantee successful clone detection, witnesses need to record source nodes' private information and certify the legitimacy of sensors based on the stored private information. In most existing clone detection protocols, the required buffer storage size depends on the network node density, i.e., sensors need a large buffer to record the exchanged information among sensors in a high-density WSN.

Present System

In this paper, besides the clone detection probability, we also consider energy consumption and memory storage in the design of clone detection protocol, i.e., an energy- and memory-efficient distributed clone detection protocol with random witness selection scheme in WSNs. Our protocol is applicable to general densely deployed multi-hop WSNs, where adversaries may compromise and clone sensor nodes to launch attacks. In previous work, we proposed an energy-efficient ring based clone detection (ERCD) protocol to achieve high clone detection probability with random witness selection, while ensuring normal network operations with satisfactory network lifetime of WSNs.

Software Requirement Specification

A Software Requirements Specification (SRS) – a requirements specification for a software system is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Nonfunctional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development lifecycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

- Business requirements describe in business terms what must be delivered or accomplished to provide value.
- Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
- Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system.

Implementation

Method1:

In the first method we will be creating frame by extending Frame class which is defined in java.awt package. Following program demonstrate the creation of a frame.

```
import java.awt.*;
public class FrameDemo1 extends Frame
{FrameDemo1()
{ setTitle("Label Frame"); setVisible(true); setSize(500,500);}
public static void main(String[] args)
{new FrameDemo1 ();}}
```

In the above program we are using three methods:

setTitle: For setting the title of the frame we will use this method. It takes String as an argument which will be the title name.

SetVisible: For making our frame visible we will use this method. This method takes Boolean value as an argument. If we are passing true then window will be visible otherwise window will not be visible.

SetSize: For setting the size of the window we will use this method. The first argument is width of the frame and second argument is height of the frame.

Method 2:

In this method we will be creating the Frame class instance for creating frame window. Following program demonstrate Method2.

```
import java.awt.*; public class FrameDemo2
{Public static void main (String[] args)
{Frame f = new Frame (); f.setTitle("My first frame"); f.setVisible(true); f.setSize(500,500);}
}
```

Testing

Implementation and Testing:

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

Implementation

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

Testing

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed.

This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

Module Testing

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

Integration Testing

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

Acceptance Testing

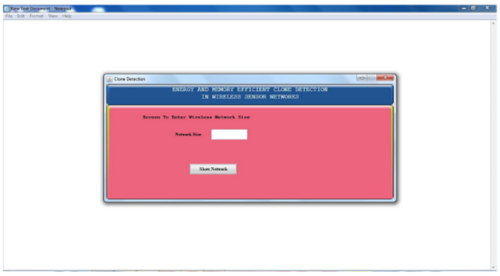
When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

Test Cases

Test Case Id	Test Case Name	Test Case Desc.	Test Steps			Test Case Status	Test Priority
			Step	Expected	Actual		
01	Show network	Verify the network size	If we enter network size as characters or if we leave empty	Then number format exception will be raised	Display the network with sensors and witnesses	High	High
02	Simulation	Test whether the sensor node verification either success or fail	If we did not select any sensor	Then verification not done	Verification successful	High	High
03	Clone Inject	Clone detection	If clone may not inject to the sensor node	Then the sensor node verification successful	Verification failed	High	High

Table:1: Test Cases

Outputs



ENERGY AND MEMORY EFFICIENT CLONE DETECTION
IN WIRELESS SENSOR NETWORKS

Screen To Enter Wireless Network Size

Network Size

Fig 1: Welcome text display

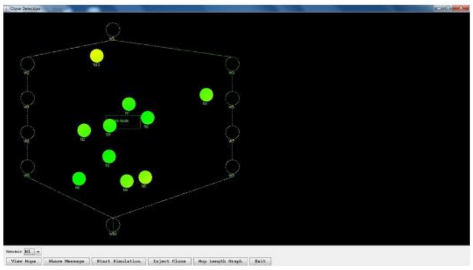
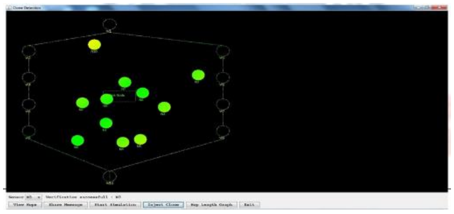


Fig 2: Network with given number of nodes



Sensor [illegible] Verification successfull : [illegible]

Fig 3: Clone Injection

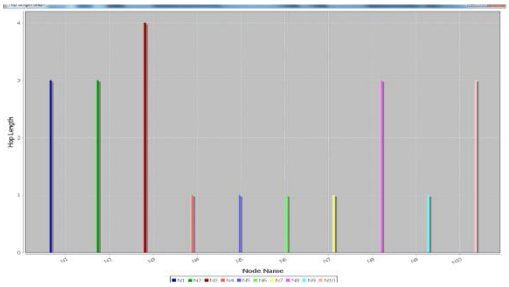


Fig 4: Hop Length graph

2. CONCLUSION

In this paper, we have proposed distributed energy-efficient clone detection protocol with random witness selection. Specifically, we have proposed ERCD protocol, which includes the witness selection and legitimacy verification stages. Both of our theoretical analysis and simulation results have demonstrated that our protocol can detect the clone attack with almost probability 1, since the witnesses of each sensor node is distributed in a ring structure which makes it easy be achieved by verification message.

In addition, our protocol can achieve better network lifetime and total energy consumption with reasonable storage capacity of data buffer. This is because we take advantage of the location information by distributing the traffic load all over WSNs, such that the energy consumption and memory storage of the sensor nodes around the sink node can be relieved and the network lifetime can be extended.

3. REFERENCES

1. Z. Zheng, A. Liu, L. X. Cai, Z. Chen, and X. Shen, "ERCD: Anenergy-efficient clone detection protocol in WSNs," in Proc. IEEEINFOCOM, Apr. 14-19, 2013, pp. 2436–2444.
2. R. Lu, X. Li, X. Liang, X. Shen, and X. Lin, "GRS: The green, reliability, and security of emerging machine to machine communications,"IEEE Commun. Mag., vol. 49, no. 4, pp. 28– 35, Apr. 2011.
3. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci,"Wireless sensor networks: A survey," Comput. Netw., vol. 38,no. 4, pp. 393–422, Mar. 2002
4. Liu, J. Ren, X. Li, Z. Chen, and X. Shen, "Design principles andimprovement of cost function based energy aware routing algorithms for wireless sensor networks," Comput. Netw., vol. 56,no. 7, pp. 1951–1967, May. 2012.
5. T. Shu, M. Krunz, and S. Liu, "Secure data collection in wirelesssensor networks using randomized dispersive routes," IEEETrans. Mobile Comput., vol. 9, no. 7, pp. 941–954, Jul. 2010.
6. P. Papadimitratos, J. Luo, and J. P. Hubaux, "A randomized countermeasure against parasitic adversaries in wireless sensornetworks," IEEE J. Sel. Areas Commun., vol. 28, no. 7,pp. 1036– 1045, Sep. 2010.
7. R. Lu, X. Lin, T. H. Luan, X. Liang, and X. Shen, "Pseudonymchanging at social spots: An effective strategy for location privacyin VANETs," IEEE Trans. Veh. Technol., vol. 61, no. 1,
8. pp. 86–96,Jan. 2012.
9. Z. M. Fadlullah, M. Fouda, N. Kato, X. Shen, and Y. Nozaki, "Anearly warning system against malicious activities for smart gridcommunications," IEEE Netw., vol. 25, no. 5, pp. 50–55, May. 2011.
10. R. Lu, X. Lin, X. Liang, and X. Shen, "A dynamic privacy-preservingkey management scheme for location based services in VANETs,"IEEE Trans. Intell. Transp. Syst., vol. 13, no. 1, pp. 127–139, Jan. 2012.
11. M. Conti, R. D. Pietro, L. Mancini, and A. Mei, "Distributed detection of clone attacks in wireless sensor networks," IEEE Trans.Dependable. Secure Comput., vol. 8, no. 5, pp. 685– 698, Sep.-Oct. 2011.