Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Key Management Scheme in Opportunistic Networks Based on Distributed Storage

Ziwen Liu
Hefei University of Technology
Jian Zhou (zhoujian@hfut.edu.cn)
Hefei University of Technology https://orcid.org/0000-0002-2972-2010
Lifan Ma
Hefei University of Technology

Research

Keywords: opportunistic network, public key management, self-organization

Posted Date: June 9th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-31282/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Abstract

Aiming at the difficulty of key distribution and identity authentication in the opportunistic network, this paper proposes a self-organized public key management scheme. This solution solves the problem of nodes in the opportunistic network through distributed storage of public keys and public key strengthening mechanisms. The problem of difficult key distribution caused by poor connectivity. Simulation experiments show that the scheme can effectively guarantee secure communication and has good anti-attack ability.

I. Introduction

Opportunistic networks are self-organized wireless networks. Moving nodes can communicate without dependence on centralized authority. Each node can communicate with other nodes within transmission range, and communicate with the nodes that are beyond transmission range in a store-carry-transmit pattern through relay nodes. The constant movement of nodes and the intermittent nature of connections make traditional key management schemes difficult to us.

Compared with traditional networks, opportunistic networks are faced with more kinds of security threats. Opportunistic networks are open, allowing foreign nodes to join the network at any time and the nodes in the network may also be disconnected from the network at any time. This openness makes opportunistic networks face security threats from within. Consequently, secure communication in opportunistic networks puts more importance on the progress in an untrusted environment, it is critical to make key management become basis of secure communication in order to solve various security issues faced by opportunistic networks. Currently, key management in opportunistic networks is done in the following ways in opportunistic networks[1].

For the first time, literature[3] proposes to apply identity-based cryptography to the opportunistic network, using the identity information of the node such as ID, mailbox, etc. as its public key. This method does not need to obtain the node certificate through CA, but requires the node to upload personal information to the private key hosting center server (PKG) to obtain its own private key. Based on this, literature[4] proposes to create a distributed private key generator PKG by selecting some of the nodes at the initial stage of the network. Literature[5] utilizes bilinear pairing to authenticate the node and key generation center, greatly enhancing the security of the transmission of information in opportunistic networks.

Literature[6] firstly puts forward the solution which makes key management by realizing distributed CA based on threshold cryptography theory. Using the (n, t) threshold cryptography scheme, we assume n child CA nodes, and only t of them are required to issue a certificate for the new node before the authentication is done. On this foundation, literature[7] proposes proactive secret sharing scheme, using n new private key shares to replace n old ones to update keys. Literature[8] uses secret sharing technology of polynomial interpolation and elliptic curve digital signature to remove third parties, further enhancing the feasibility of threshold-based cryptography methods.

Literature[9] firstly proposes a fully distributed self-organizing public key management scheme of authenticating in the form of a certificate chain, which originates from PGP algorithm. Node issues certificates for the others and collects certificates from them, each node will form a certificate base to store certificates they have issued and collected. When two nodes communicate, they merge each other's certificate bases, from which a path of certificates being verified in turn should be found to form a certificate chain to verify the legitimacy of the two nodes. Literature[10] uses the certificate-chain method, a small number of adjacent nodes are selected to establish trust and issue certificates at the initial stage of the network, which effectively improves the robustness of the algorithm. Literature[11] proposes to enlarge the covering area of virtual CA by the certificate chain and utilize trusted nodes as initial nodes of the chain, in order to enhance security. Literature[12] proposes a certificate chain scheme which is based on the unit of trust group. Trusted nodes form into a trust group and generate a public/private key pair of the group. Any group member are permitted to use this key pair to issue certificates and authenticate.

From our perspective, the methods that are based on trusted third party(namely CA,PKG) violate the characteristic of opportunistic networks which is being self-organized. In addition, selecting nodes to undertake the task of authentication or issuing certificates is contrary to the equality of nodes and openness of the network. Self-organizing public key management is an ideal choice to work out a key allocation scheme in opportunistic networks because trusted third party and complex authentication process are not required. An important problem in the method of certificate chain is how to use the trust model as the basis for measurement, that is, how to ensure that the nodes in the certificate chain are honest. The security of the certificate chain depends to a large extent on the selection of the trust model, which brings a certain security risk to the certificate chain model. And complex computing and storage costs are also problems that still exist in the certificate chain model.

This paper believes that due to the characteristic of the opportunistic network’s being self-organized, if we adopt the measures of authorization and authentication from self-organized key management, it will bring inevitable security risks. This paper proposes distributed storage of the public key of the nodes, and to defend attacks from inside the network by reinforcing public key through the use of it during the message transmission.

Ii. Public Key Management Scheme Of Distributed Storage

The node independently generates its own public and private keys and diffuses the public key to all nodes in the network as quickly as possible. The rapid proliferation of public key not only enhances the availability of secure communications but also reduces the likelihood of ID attacks. the use of public keys in communication enhances the trust of the public key in opportunistic networks, and the impact of malicious node attacks can be resisted through voting mechanisms.

1. *Security Threats Faced by Self-Organized Public Key Management*

Based on CA or identity cryptography, the legitimacy of the public key can be fully verified. Unlike this, it is much more difficult to prove the legitimacy of the node identity in self-organized public key management.

In addition that inadequate and untrustworthy authentication carries its own risks, the transfer of trust to a certain extent also lack rationality on this basis. More specifically, it will be difficult to prove whether the binding of node identity and public key is legal, creating opportunities which can be made use of to conduct internal attacks. And identity-based attacks such as node phishing and tampering with public keys of other nodes to steal messages are the main problems faced by self-organized public key management scheme.

The lack of certificates issued by authorities makes it difficult to verify the credibility of the node's identity, and the certificate issued by the third-party node involves its own trust in the behavior of the third-party node, thus it is one of the burning issues which needs to be solved in opportunistic networks that how to make secure message transmission in an untrustworthy environment. This paper holds that the binding of node identity to the public key requires more recognition than verification, when stored in a certain number of nodes, the binding of node identity and public key will have a certain degree of credibility, so as to resist the attack based on node identity.

1. *The Validity of Public Key*

The validity of the public key is not only the identification of the legitimacy of a node's public key, but also represents the degree of trust of other node's binding between its identity and public key. The validity of a node's public key identified by each node is different, representing the different degrees of trust in the public key of the node. With the use of the public key in the network, the node's trust in the public key is also enhanced, and the upper limit on the validity of the public key is promoted.

When a new node goes live or queries from a neighbor to an unsaved public key Pk_i, the public key Pk_i of the node N_i is saved as the initial value Val_0 of the validity.The selection of Val_0 is a very important parameter in this paper, proper initial value indicates that the binding of ID_i(the identity of N_i) to Pk_i is not sufficiently trusted by the node. Thus it is necessary to refresh frequently to verify the validity of the binding. At the same time, the binding is required not to be expired, or it will affect storage and use of Pk_i in the network.

The method of calculating Val_0 used in this paper is shown in Eqs. (1) :

$$\frac{1}{Val_0} = \frac{1}{2} Fre \left(1 + \frac{\alpha \times Rec \times 2\mathrm{h}op}{Net}\right)$$

1

where Fre is the average contact frequency, Rec is the message reception frequency, $\mathrm{h}op$ is the average number of nodes that assist in forwarding a message which is received or transferred,$2\mathrm{h}op$ is the average routing hops of a communication,Net is the total number of nodes in the network and α is adjusted according to the marginalization degree of the node. When a node receives or forwards messages more frequently and the average routing hops for a message becomes larger. It indicates that

the node is more likely to be aware in unit time that the public key of another node is still in use in the message transmission, and Val_0 is shorter.

The validity of the public key will be extended with the use of this key. When a node N_i is stored and the validity of its public key Pk_i has exceeded threshold η(which has been set in the simulation), the correctness of this Pk_i will not be questioned despite its inconsistency with the Pk_i stored by other nodes. The validity of the public key is updated in two ways:

1. When a message is transmitted, the owner of the public key uses the corresponding private key for signature. If the local node just assists in the transmission of the message or is the destination node of this transmission, the validity of the public key will be reset and its upper limit will be increased. In a certain period of time, the upper limit of the validity of the public key cannot be increased repeatedly.
2. When the public key is in the middle of its validation, the node will actively seek an update. When the node encounters a neighbor, if the validity of the public key stored by the neighbor node is longer than that of its own, the validity of the public key will be reset but its upper limit will not be increased.

Definition 1

Freeze Period: When two nodes meet, node can assume that the public key claimed by node is trustworthy for a long time,and the public key of locally stored by will be frozen for a period of time. It will not be asked or changed even when it is ambiguous with other nodes.

1. *Network Initialization and New Node Going Live*

When it is at the initial stage of the network or a new node goes live, this new node n_i will autonomously generate its own public key Pk_i and private key Prk_i as well as an on-line notification. The notification adopts a fixed message header format, attaching the node's ID and Pk_i along with Prk_i signature. When the new node encounters its first neighbor in the network, it will copy the public key of each node in this neighbor node's public key list, and all these public keys' validity is Val_0. According to the size of the network and other basic parameters, we calculate the valid time of the on-line notification. In this paper, the valid time is several times the message flood time delay. In the notification time, nodes will receive multiple same on-line notifications on average, and when there are inconsistencies in identifying Pk_i in multiple on-line notifications, it can be discovered and the voting mechanism described below can be used in a timely manner.

When the network is under severe attack, the voting mechanism cannot fully correct errors from identity attacks. At this point, the nodes which cannot define the public key Pk_i of n_i will not record Pk_i in a following period of time(denoted by ξ) as well as receive or broadcast the on-line notification of n_i. If the network attack persists for a long time, it will be continuously difficult to determine the Pk_i for this part of the nodes, and the freeze period plays an important role in solving this problem. The nodes that meet n_i

should make n_i's public key Pk_i into a freezing period, and the nodes that directly meet n_i will not be affected by the attack in terms of recording Pk_i, also to a certain extent ensuring the availability of Pk_i. After a ξ time, the network will accumulate a certain number of nodes that have met n_i directly. The proliferation of Pk_i through this part of the nodes will effectively prevent the public key from being tampered with and increase the likelihood of correction of the voting mechanism.

1. *Voting Mechanism*

The voting mechanism is that when the node is in doubt about the correctness of a certain public key, it uses the inquiry mechanism to the nodes in the network. The source node n_1 sends several voting packets along different paths to the target node dst that is ambiguous lying on the public key, and all intermediate nodes attach the public key of dst in their own local public key list in into the voting packets. Dst also attaches its own public key Pk_{dst} and sends it back to the source node n_1 when it receives the packet. If the nodes that sign Pk_{dst} in all voting packets account for more than voting threshold vt, then Pk_{dst} is considered to be trusted by n_1. In the following two cases, node n_1 will seek a vote:

1. n_1 has received multiple versions of new node on-line notification.
2. n_1 and its neighbors have ambiguity about the public key of a certain node in the network, and the validity of this node stored by n_1 is not longer than η.

Before node n_1 seek a vote, the number of voting packets is defined by the scale of the network and related data are prepared in advance. Vot is voting data marker, id_{dst} is the identity of dst, ts is the time stamp at which the voting packet was generated. When n_1 meets another node in the network, if this node is appropriate to be the next hop of the voting packet and it has not received the packet, then n_1 prepares to transmit the packet to this node(henceforth referred to as n_2).n_1 records the identity information of n_2 as id_{n_2}, and signs Pk_{dst},id_{n_1},id_{n_3} as$\mathrm{Prk}_{n_2}(\mathrm{Pk}_{dst}, \mathrm{id}_{n_1}, \mathrm{id}_{n_3})$ using its private key Prk_{n_1}. When n_2 receives the packet, it also records the identity information of n_1 as id_{n_1} and checks which node that n_1 is inquiring about. According to the public key list of n_2, we search for Pk_{dst} and use Prk_{n_2} to sign Pk_{dst}, id_{n_1}, id_{n_3} as $\mathrm{Prk}_{n_2}(\mathrm{Pk}_{dst}, \mathrm{id}_{n_1}, \mathrm{id}_{n_3})$ when n_2 meets next hop n_3, after which n_2 transmits all the data in the packet(including the data n_1 has transmitted to n_2) to n_3. This pattern of behavior will be conducted repeatedly until the voting packet has finally transmitted to node dst. This transmission format is called Chained Transmission Mode, which is to prevent voting packets from being tampered with or selectively deleted by intermediate nodes or dst. Figure 1 describes the data format that uses Chained Transmission Mode:

When dst receives the voting packet, it records likewise the identity information of last node n_3 as id_{n_3} and generates ts' which indicates the time of its receiving the packet, attaching its own public key Pk_{dst} into the packet. What's more,dst encrypts Pk_{dst}, id_{n_3}, ts' with Pk_{n_1} to $\mathrm{Pk}_{n_1}(\mathrm{Pk}_{dst}, \mathrm{id}_{n_3}, ts')$ and sends all the voting data to source node n_1. n_1 collects all the voting data and picks up Pk_{dst} from the nodes

involved in the packets. If the highest percentage of Pk_{dst} is higher than threshold vt(75% in the simulation), it can be considered to result from network error or a few malicious nodes' diffusing the wrong public key of dst. At this point,n_1 will record Pk_{dst}. If none of the Pk_{dst}s recorded in the voting data account for more than vt, then it is difficult to define the true public key of dst. And node u will stop recording the information of Pk_{dst} for a following period of time ξ.

1. *Node Interaction and Public Key Diffusion*

Each node generates their own public key list to store ID, the public key, the validity and online timer of other nodes in the network, and the list is above board so any node is permitted to check. A typical public key list is as bellow:

1. Public Key List

Alive Nodes: 7				
Online Notifica-tion	**ID**	**Public Key**	**Online Timer**	
	n1	Pkn1	3:50	
	n8	Pkn8	5:00	
Soon to Expire	**ID**	**Public Key**	**Expire Time**	**validity Duration**
	n2	Pkn2	13:35	10:00
	n5	Pkn5	13:21	7:00
In Use	**ID**	**Public Key**	**Expire Time**	**validity Duration**
	n3	Pkn3	15:32(Frozen before 17:00)	10:00
	n4	Pkn4	14:05	8:00
	n7	Pkn7	14:34	6:00
Invalid	**ID**	**Public Key**	**Invalidity Time**	
	n6	Pkn6	12:30	

When node u meets neighbor v, u updates its public key list according to the following rules:

1. When node u receives an on-line notification from node N_i and the timer for that notification is still valid, node v is reminded to update the public key list.
2. If there is no notification of the nodes in the public key list of node u coming online and no public key is about to expire, there is no need to update the public key list actively.
3. If node v receives an online notification from a new node N_i or there exists a public key of some node in the public key list of node u that is about to expire, the public key list will be updated through

node v. First u receives the notification of node v and checks if it has received this notification before, and if not, adds N_i's ID and public key Pk_i to the list, while inheriting the timer in v that records N_i's on-line notification. If node u has received the notification before, compare whether the two Pk_is in the notification are consistent and use the voting mechanism if not.

4. Check the validity of Pk_j which is about to expire in the local public key list from the list of node v. First to verify that if the records of Pk_js in u, v are consistent. If they are consistent and the validity recorded in v is longer than that in local record, the list is updated to a shorter one between the validity recorded in v and the validity reset from the current moment, if not consistent, determine whether the total length of the validity of the local record Pk_j is greater than the threshold η, and enter the voting mechanism if less than η.
5. Compare the number of u's and v's public key list entries, if the entries are inconsistent, find the ID and public key of the missing node, its validity is updated to Val_0.

Iii. Simulation And Performance Evaluation

This paper carried out the simulation on ONE platform to analyze the availability and security of this method. The simulation uses a node motion model of Working Day Movement, setting up the scene similar to literature [13] as below:

1. Simulation Setting

Num of node	**200**
Size of map	3800 × 3000 m^2
Transmission range	10 m
Simulation duration	43200 s
Walking speed	0.8–1.4 m/s
Car speed	8–12 m/s
Night trip rate	0.5
Office pause coefficient	0.5
η	1h
ξ	30 min

Fig. 2 compares the average success rate of node authentication in this method with the traditional certificate chain method. As can be seen from Fig. 2, the nodes in this method can quickly achieve a high success rate of mutual authentication, because there is no need of warm-up time. Whether it is the initial

node of the network or the node that later joins the network, both of them can be quickly authenticated. While the traditional certificate chain method requires a certain amount of time to complete the certificate link, faced with the problem that when the new node joins the network, it is unable to communicate securely during this period of time. In terms of network availability, it is very important for nodes to be able to quickly integrate into the network and ensure security when they enter the network. This method is more practical in this light.

In this method, after the success of the authentication of majority of the nodes, there will be a period of volatility in the success rate of authentication, which is caused by the node's generally short validity and failure of resetting the node before the end of the validity. Meanwhile, in the experiment we found that frequent updates of public key list increase the computing cost of the node. With the simulation time increased, the validity of the public key of each node increases significantly, and the network operates at a lower power consumption.

The security analysis of this scheme is one of the key contents of the simulation experiment. We randomly selected several groups of malicious nodes where the number of nodes differs from each other. What's more, these nodes collude with each other. After this, we began identity attack aiming for 10 random nodes in the network. During the attack, the malicious nodes constructed the same false public key, and intercepted voting packets on the basis of routing protocol. Figure 3 shows the different effects of the attack in terms of different time to launch the attack after the target nodes go online. If more than half of the nodes in the network cannot judge or record whether the public key of the target node is wrong, we consider the attack to be effective. The simulation have been carried out 10 times and an average value was calculated.

Fig. 3 shows that the success rate of identity attacks is sensitive to the time the node has survived. When nodes have survived in the network for 10 to 15 minutes, even increasing the number of malicious nodes, it is difficult to operate effective attacks. Therefore, nodes that live relatively longer in the network are highly secure, and these nodes often play an important role in the network and are more valuable to be attacked.

Although the newly launched nodes are not of much value to be attacked, we remain concerned about the possibility of them being attacked. Figure 4 shows the effect of malicious nodes attacking new nodes when they go live according to the trend of time, wherein the vertical axis is the proportion of the public key of the target node being successfully diffused. Because of the existence of the attacker nodes, some nodes in the network may receive inconsistent node on-line notifications and cannot determine the correct public key. The public key will not be stored or forwarded in the following $\xi = 30$ minutes, and the number of nodes that meet directly with the target node cumulatively increases during this time, greatly promoting the subsequent proliferation of the public key. This simulation experiment relies heavily on the preparation made by the malicious nodes in advance, and the attack effect is under the situation that all malicious nodes can collude. This paper does not study the problem of whether the malicious node can collude effectively.

Considering that opportunistic networks are still partly used in low-power devices and node-sparse networks, we have calculated the computational cost of maintaining this method. Figure 5 shows the probability that nodes will need to exchange the list of public keys on average in each encounter as the simulation time increases and the validity of public key is extended. In the simulation experiment shown in Fig. 5, there are no new nodes to join and exit the network. As can be seen from the simulation results, the nodes need to frequently exchange the public key list at the initial stage of the network, and the validity of the node's public key continuously increases with communication between nodes, gradually reducing the cost of maintaining the public key list. After the network runs stable, the node maintains the public key list at a very low cost. Compared with maintaining the local certificate library and finding a certificate link in the certificate library required by the certificate chain, the cost of storing and calculating in this method is lower.

Iv. Conclusion

In this paper, we put forward a set of key management scheme in the opportunistic network, which advocates the efficient diffusion and distributed storage of the node's public key as well as strengthens the trust of the public key through the use of public key. In addition, our scheme proposes to reduce security risks brought by network error and malicious nodes through voting and other mechanisms. We have fully simulated this scheme in the aspects of availability, security and power consumption. The results show that our scheme has a higher operating efficiency, which can ensure the safety and security of the network when it is under attack. There is no absolute security in opportunistic networks, and our follow-up work will continue to explore secure and effective public key authentication methods to further improve the security of communication in opportunistic networks.

Abbreviations

Val : Validity

Pk_{n_i} : The public key of n_i

Prk_{n_i} : The private key of n_i

id_{n_i}: The identity of n_i

ts : Time-stamp

Declarations

Competing Interests

The authors that declare no competing interests.

Funding

The work was supported by the National Natural Science Foundation of China under Grant No.60873194 and supported in part by the National Natural Science Foundation of China under Grant U1836102.

Availability of data and materials

The data sets supporting the results of this article are included within the article. All other data supporting the findings of this study are available from the corresponding author upon request.

Authors' contributions

Z.L. conceived and designed the experiments and contributed to the development of experimental methods as well as collected the data. L.M. analyzed the data and wrote the manuscript. J.Z. agreed the submission of the manuscript. All authors provided suggestions for the experiments, discussed the results and contributed to the editing of the manuscript.

References

1. Biagioni E, Giordano S, Dobre C. Ad Hoc and Sensor Networks[J]. IEEE Communications Magazine, 2017, 55(1): 166-167.
2. DENG H, MUKHERJEE A, AGRAWAL D P. Threshold and Identity Based Key Management and Authentication for Wireless Ad Hoc Networks[C]//IEEE. IEEE International Conferences on Information Technology:Coding and Computer(ITCC'04), April5-7, 2004, LasVegas, Nevada. New Jersey: IEEE, 2004: 107-110.
3. Seth A, Keshav S. Practical Security for Disconnected Nodes[C]. Secure Network Protocols. 2005.
4. Aram Khalili, Jonathan Katz, William Arbaugh. Towards security solutions for truly ad-hoc networks. IEEE Workshop on Security and Assurance in Ad Hoc Networks, in conjunction with the 2003 International Symposium on Applications and the Internet, Orlando, FL, January 28, 2003, pp.1511-1515.
5. Li H X, Pang L J, Wang Y M. Key management scheme without secure channel for ad hoc networks [J]. Journal on Communications, 2010, 31(1):112-117.
6. Lidong Zhou, Zygmunt J Haas.Securing ad hoc networks[J] .IEEE Networks Special Issue on Network Security, 1999, 13(6) :24-30.
7. Herzberg A, Jarecki S, Krawczyk H, et al. Proactive Secret Sharing Or: How to Cope With Perpetual Leakage[M]. Advances in Cryptology — CRYPT0' 95. 2010:339-352.
8. Li J, Liu Q, Wang A, et Study on the distributed key generation algorithm in mobile Ad-hoc networks[C]. Second Pacific-Asia Conference on Circuits,communications and System. 2010:438-441.

9. Capkun S, Buttyán L, Hubaux J P. Self-Organized Public-Key Management for Mobile Ad Hoc network[J]. IEEE Transactions on Mobile Computing, 2003, 2(1):52-64.
10. Dahshan H, Irvine A robust self-organized public key management for mobile ad hoc networks[J]. Security & Communication Networks, 2010, 3(1):16-30.
11. Yi S, Kravets Composite Key Management for Ad Hoc Networks[C]. IEEE MobiQuitous. 2004: 52-61.
12. Chang C P, Lin J C, Lai Trust-group-based authentication services for mobile ad hoc networks[C]. IEEE 2006 1st International Symposium on Wireless Pervasive Computing, 2006: 4.
13. Ekman F, Keränen A, Karvo J, et al. Working Day movement model. Proceedings of the 1st ACM SIGMOBILE workshop on Mobility New York: ACM Press, 2008:187-198.
14. On-demand key management based on social attribute for opportunistic networks[J]. Journal on Communications, 2013, 34(12): 93-99.

Figures

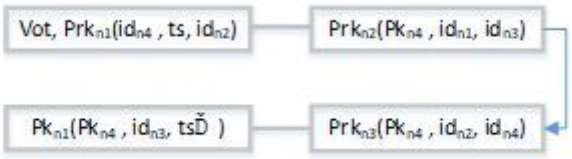

Figure 1

Chained transmission mode

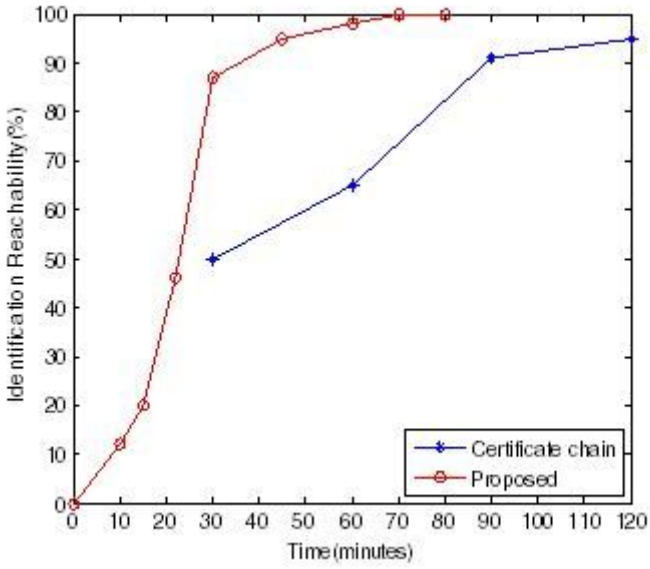

Figure 2

The comparison of success ratio

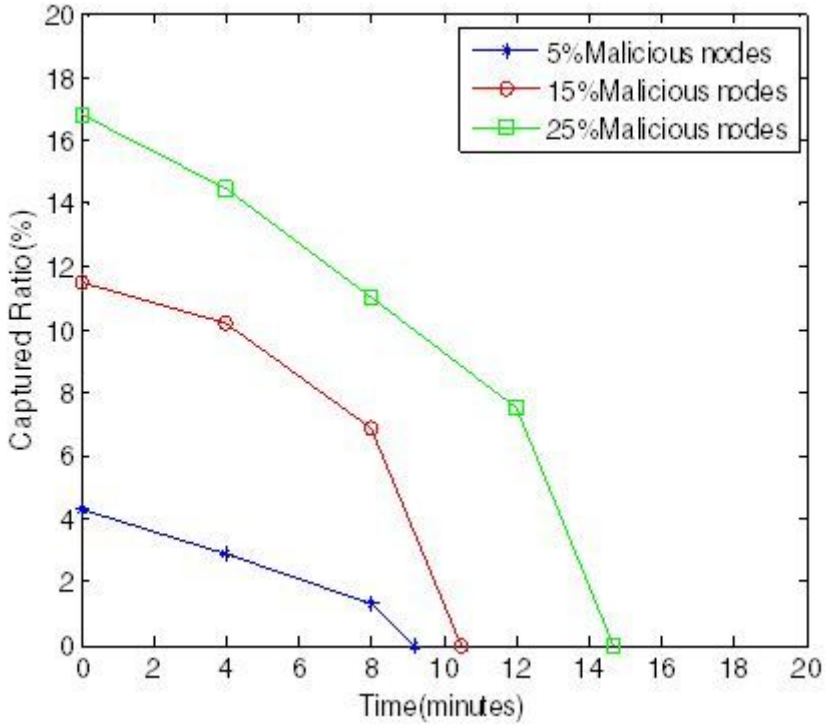

Figure 3

Impact of attacks at different times

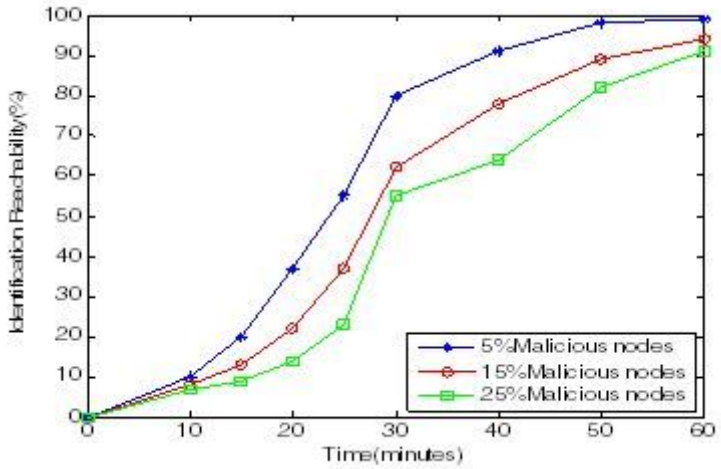

Figure 4

The change of attack impact over time

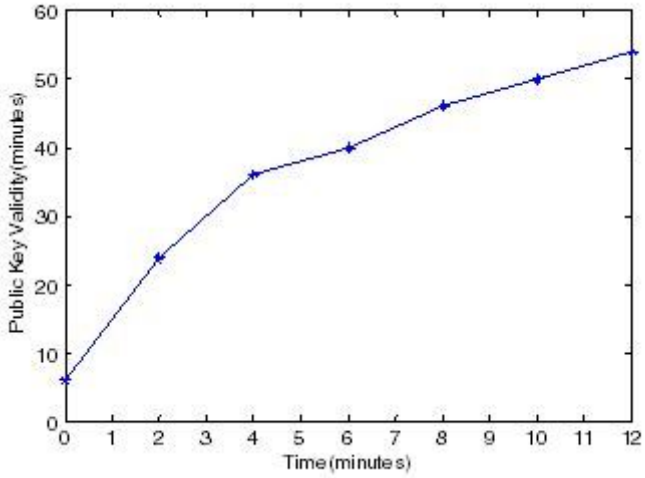

Figure 5

The change of node's average validity along with survival time