ESKA: A Highly Reliable Authentication Protocol Based-on One-way Key Chain for WSN Broadcast

Qianping Wang, Ruoyu Li, Liangli Lai, Lei Kong
School of Computer
China University of Mining and Technology
Xuzhou 221116, China
Email: qpwang@cumt.edu.cn, vincentlry@126.com

Submitted: Mar. 30, 2013 *Accepted: July 15, 2013* *Published: Sep. 05, 2013*

Abstract- The authentication of broadcast sources is one of the most important research issues in the area of network security. To confirm that the control information received is from the genuine management nodes, the message broadcasted from management nodes must be verifiable for the working nodes. In this paper, a highly reliable WSN broadcast authentication protocol based-on one-way key chain, namely, the ESKA, is presented. For the purpose of comparison, a group of simulations are conducted. The results have demonstrated that compared with μTESLA, the presented method is of lower authentication delay

and higher authentication rate in large broadcast data traffic transmission, ***meanwhile reducing considerable communication overheads in those applications.***

Index terms*:* Broadcast Authentication Protocol, μTESLA, ESKA, High Reliability, One-way Key Chain.

I. INTRODUCTION

Sensor nodes are usually deployed in complex environments, and they are prone to be affected by external factors including human factors [1]. Meanwhile, nodes failure and many other factors may lead to the change of network topology [2]. Any kinds of secure issues of WSN may undermine the reliability of WSN applications, e.g., the commercial wireless security network in districts, monitoring the WSN deployment in enemy-occupied area for military use [3]. Traditional security mechanisms and protocols do not take the nods' characteristic into consideration, therefore they cannot be applied in WSN directly. Taking into account that the nodes in WSN are limited in computing speed, power supply, communication and message storage, it is demanded to design a robust security mechanism which can provide appropriate services and protect the WSN from malicious attack [4, 5].

Authentication is one of the principal security services in WSN, which aims to certify identity of nodes and correctness of messages from the source nodes. Many security services are based on the authentication issues [6]. In WSN, there are two kinds of authentication issues, including P2P authentication and broadcast authentication. P2P authentication is designed for certifying the communication between two points. Being different, broadcast authentication is an authentication for one node to multi-nodes. The traditional P2P authentication uses symmetric encryption technology to certify the message source: both sides communicate with each other by using the shared key to encrypt the messages. The source authentication of broadcast communicating cannot use the same mechanism, as any nodes within the network can imitate

the source node and send ciphertext [7]. Once any potential attacker obtained the shared key from certain node, the security of the broadcast authentication mechanism would be broken. By contrast, it would be better to choose the asymmetric encryption, which is a digital signature algorithm based on the public key mechanism. The sensor nodes are limited in computing speed, power supply, communication and message storage, traditional digital signature algorithms based on public key mechanism is not suitable for WSN, hence a novel broadcast authentication mechanism for WSN is demanded.

II. Related Work

There are many research challenges for WSN [8]. A suitable security protection mechanism is demanded to ensure that the communication in network. One of the many requirements is security according to the characteristic of WSN [9, 10, 11]. The protocol SPINS is one of WSN security framework. For the limited resources and the optimal architecture of wireless communication, there are two secure modules of SPINS, namely, the SNEP and the μTESLA [12]. The former one outlines a very important basic security principle: data confidentiality, double sides' data authentication, the freshness of the data and P2P authentication. The μTESLA could use broadcast authentication in WSN with strictly limited resources.

In Figure 1, the μTESLA broadcast authentication process is illustrated. The broadcast nodes broadcast packages P_1~P_6 and release 4 keys K_{i-1}~K_{i+2} in 4 key cycles. The nodes will check packages' time information after they receive packages P_1 and P_2, then determine that two packages' broadcast keys do not release, So P_1 and P_2 will be stored by sensors. The sensor nodes save the time information when the key will be released, and will receive the key information at the right time. After receiving key K_i, the nodes will calculate $F(K_i)$ and check that whether $F(K_i)$ is consistent with K_{i-1}. If not, the nodes will drop the key. Otherwise, the key will be considered to be legal. After

receiving the legal key, the nodes will use Ki to authenticate P_1 and P_2 receiving in $[T_i, T_i+T_{int}]$ according to the time scale.

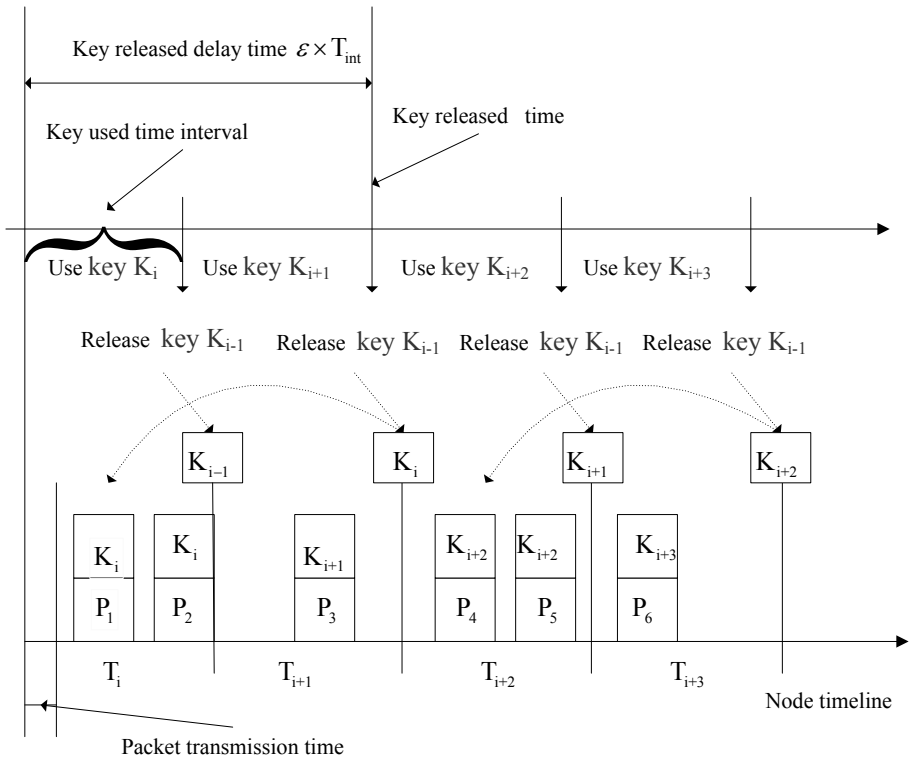

Figure 1. Broadcast authentication process of μTESLA

If the node dropped the released key K_{i+1}, it can certify P_3 after receiving the next broadcast key K_{i+2} that is the key certifying $[T_{i+2}, T_{i+2}+T_{int}]$. Due to the loss of K_{i+1}, the nodes can judge the key by comparing that if $F^{(2)}(K_{i+2})$ and K_i is equal. If it is legal, then it needs to calculate $K_{i+1}= F^{(1)}(K_{i+2})$, and use K_{i+1} to certify P_3, use K_{i+2} to certify P_4 and P_5.

The μTESLA tackles some problems in broadcast authentication such as:

1) Shares the network key,
2) Generates one-way of the key chain,
3) Releases the lost key packages,
4) Delay key's releasing,
5) Authenticate and initiate the key.

The drawback of the μTESLA lies in that the authentication delay may bring about DoS attack in the cycle of distributing parameters.

III. The Broadcast Authentication Scheme ESKA

Traditional authentication schemes usually adopt asymmetric key to carry out authentication, such as RSA, DSA and so on, but those schemes require large memory for message storage, which may cause more energy consuming [13]. As a result, those schemes are not suitable for WSN [14]. Although in some relative works, it was suggested that the elliptic curve key algorithm which has shorter key message and less calculating process can be used for WSN to carry out authentication, but the memory and energy consuming is still too large for real WSN applications [15, 16]. This paper adopts symmetric encryption algorithm to carry out broadcast authentication in WSN [17]. ESKA is based on the following assumptions [18]:

1) The base station is of strong calculating capability, large memory and well protected,
2) The nodes of WSN can communicate with the base station either directly or indirectly,
3) Before deployed in WSN, the nodes have stored the Hash algorithm codes, initial key and other information.

a. System initialization

The packages' form used by base station to carry out broadcast authentication is as table 1. Hd represents the header of signature package, which has source address, package's length, and destination address.

Table 1: Signature packages' form of base station

Form	Explanation
Src	source address
Len	package's length
Dst	destination address
Cmd	current packet type
Rnd	signature round number
Ctr	Timer
Data	valid data
Mac	Signature

a.i The generation of key chain

The base station saves shared one-way Hash function by the whole network before the network is deployed, then generates the key chain with the key generating algorithm [19]. The generation of key chain is based on one-way function, MD5 is used. MD5 is a classical Hash algorithm, it can deal with any message and then get the 128bit key. It is suitable for WSN in security and calculating cost.

The first key $K_0^{'}$ of the key chain is generated randomly, then the p+1 key $\{K_0^{'}, K_1^{'}, K_2^{'}, ...K_p^{'}\}$ by the following key generating process.

$$\begin{aligned} K_1^{'} &= F^{(1)}(K_0^{'}) \\ K_2^{'} &= F^{(2)}(K_0^{'}) \\ K_3^{'} &= F^{(3)}(K_0^{'}) \qquad \text{..... (1)} \\ &\text{.....} \\ K_p^{'} &= F^{(p)}(K_0^{'}) \end{aligned}$$

The process of the base station using the key chain is opposite process of the generating of the key chain, which can be written down $L=\{K_0, K_1, K_2, \ldots, K_p\}$, where $K_0=F(K_1)$.

a.ii The release of initial key

At the beginning, nodes need to join in WSN. First they must register in base station, and get the necessary security information. The process is as follow:

$$A \to BS : ID_A \parallel N_A \parallel \mathrm{Re}q \parallel MAC(K_{au}, ID_A \parallel N_A \parallel \mathrm{Re}q) \quad \text{..... (2)}$$

$$BS \to A : (K_0 \parallel \mathrm{ID}_A)K_{en}, MAC\{K_{au}, N_A \parallel (K_0 \parallel \mathrm{ID}_A)\}K_{en} \quad \text{..... (3)}$$

A is the sensor node, BS is the base station, N_A is a random number that can carry out strong refresh authentication. Req is the package that requests to join in the WSN. BS once receives packages with Req, it must copy the node's information. K_{au} is the authentication key between A and S. K_0 is the initial key, and also the first key information of the Hash chain. K_{au} is the authentication key between A and BS, Ken is the encryption key between A and BS when they carry out communication. They both generate the shared key information:

$$K_{en} = F^{(1)}(K_{start}) \quad \text{..... (4)}$$

$$K_{au} = F^{(2)}(K_{start}) \quad \text{..... (5)}$$

With the above steps, the sensor node receives the first key K_0 in one-way key chain. Then when BS releases the K_1, it can calculate $F(K_1)$ to distinguish the released key by checking whether $F(K_1)$ is equal to K_0. The base station signature and the sensors authentication will be safe.

b. Broadcast signature and authentication

The base station sends control information to sensor nodes in the way of broadcasting packages [20]. The ESKA is based on one-way key chain broadcast authentication algorithm, which core part is to solve the base station signature and nodes certifying.

b.i The BS signature and the key transmit

The base station sends broadcast packages M_{A1} to sensor A:

$$M_{A1} = \{H_d \parallel Cmd \parallel \mathrm{Rnd} \parallel Ctr \parallel Data \parallel Mac_{A1}\} \quad \text{..... (6)}$$

$$Mac_{A1} = Mac(K_j, \parallel H_d \parallel Cmd \parallel \mathrm{Rnd} \parallel \mathrm{Ctr} \parallel \mathrm{Data}) \quad \text{..... (7)}$$

The Figure 2 shows the process of sending the base station's signature packages. The base station node and sensor nodes have the same buffer packet's threshold R. When the size of data packet reach the threshold, the base station will consider that it has already reached the max number of buffer packets, then it will broadcast the key packets regardless of whether the time slot is T.

The process of the algorithm of broadcasting key packets:

Step 1: The base station checks that whether there exist packets to be sent. If exist, it will send the packets, then generate MAC_i with encryption.

$$Msg = \{M_i \parallel MAC_i\} \quad ..... \quad (8)$$

$$M_i = \{H_d \parallel Cmd \parallel Rnd \parallel Ctr \parallel Data\} \quad ..... \quad (9)$$

After processing, the base station will broadcast it to sensor nodes, the counter will increase 1. If not, it will turn to step 3;

Step 2: The base station judges whether the signature packet's counter is greater than threshold R, if it has reached the threshold R, which means that the node has reached its limit for caching the signature packet and can no longer cache signature package. At this time, base station node will broadcast key packets to help the sensor nodes certify the received signature data packet and then set counter Num and timer zero. Turn to step 1. Or, turn to step 3;

Step 3: Check whether the timer has reached the moment key is released. If not, turn to step 1;

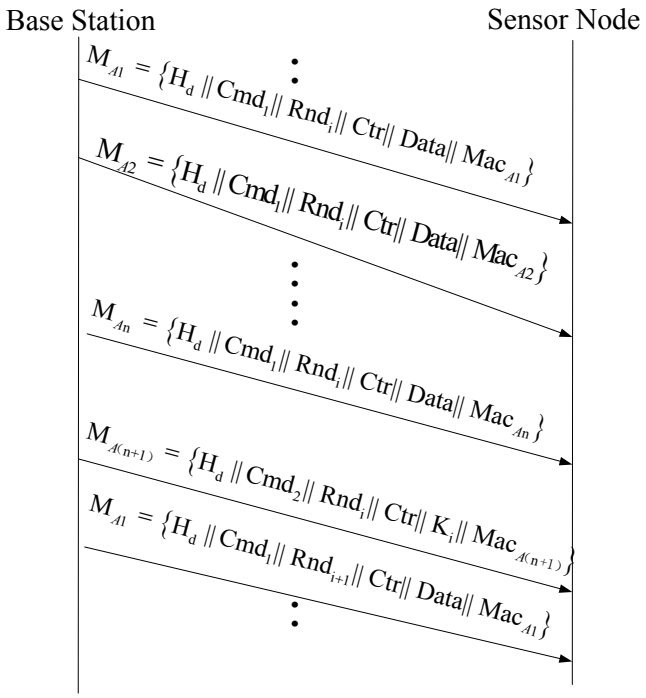

Figure 2. Signature packets and key packets from base station

Step 4: If it has reached time T, the base station node will check whether the signature packet counter Num is greater than zero. If, it means that it is not broadcasting the signature packet in last period. It is no significant to broadcast key packet. At this time, the base station node will not release the key packet information, but set the timer zero (re-timing), then postpone the release of next authentication key;

Step 5: If Num is greater than zero, which means that the base station node broadcasts the signature packet in the last period. At this time, base station broadcasts key packet, and clears the signature packet counter Num and the timer T, turn to Step 1.

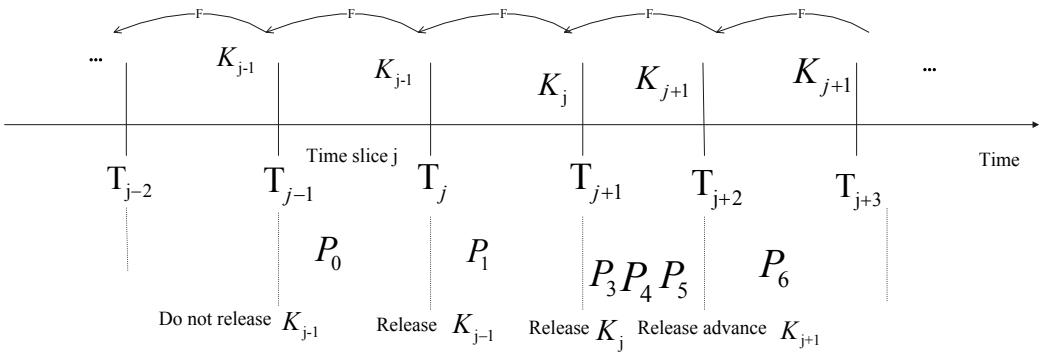

Figure 3. Process of broadcasting signature packets and key distribution

Figure 3 is the process of broadcasting signature packets and sending key. At T_{j+1} moment, sensor nodes receive the key information K_j. It will calculate $F(K_j)$ first, then compared with K_{j-1}. If equal, the key is a legitimate key, otherwise drop the key package. After receiving valid key K_j, nodes will find the cached data packets which use signature of this key information according to Rnd in the packet. The entire life cycle of the base station is divided into some time slices. In each slice the base station will use the key in one-way key chain to sign data packet in the corresponding period. Generating the sequence of key and releasing the key chain are opposite.

b.ii Packet caching and digital signature authentication

After node receiving the signature packet from the base station, it will deal with data packets differently according to Cmd and Rnd. When generating the packet signature, it will use symmetric encryption algorithm RC5. RC5 is a simple and efficient algorithm, and it is suitable for hardware and software implementations, it is also a customizable encryption algorithm. The sensor nodes can choose different parameters to carry out cryptographic operations, which is very flexible.

When receiving signature packet $Packet_1$, the node must firstly check whether the packet is the base station broadcast packets or forged by the attacker. Cmd and Rnd from the packet and then checks whether the Cmd is in the set of

Cmd commands, Rnd is in reasonable range. If any one is not in the correct range, it indicates that the packet is forged or maybe mistakes occurred in sending, followed by dropping of the packet. Otherwise, the node will continue to check wheterh Ctr is correct or not.

It can check the packets' freshness by counter Ctr. Ctr corresponds to an increment counter in the base station node. The counter will plus 1 when sending a data packet. The sensor nodes should check whether the Ctr is greater than Ctr' the last received packet. If $Ctr > Ctr'$ is wrong, drop the packet. Then the base station judges data packet type of Cmd, and fulfills corresponding operations.

If the packet received by the base station node is key releasing packet, base station node will begin the certification process of the packet. The sensor node will take out Rnd_i and K_j from key released package, and finds the data packet include Rnd_i. Each data packet collected in the data package set will be certified using the K_i. Then the sensor node certifies Rnd_{i-1}.

The base station node and sensor node have same packet buffer threshold R, all the sensor nodes will not receive more data than the threshold R package. If the number of the received packet is more than this number, then the node can judge that they have already suffered DOS attack.

c. The refresh of the key chain and new node joining in

c.i The refresh of one-way key chain

In initialization, the base station node generates a key chain $L = \{K_0, K_1, K_2, ...K_p\}$. If the key chain is depleted in the life cycle of the base station, broadcast signature and certification process will not continue, so the base station need to generate new key chain to continue this signature certification process.

Base station generates a new key chain which can not be used immediately. The sensor nodes will check the received key information and judge its

legality. If $F(K_{i+1})$ is not equal to K_i, the sensor node will consider that this is a fake key release packet and drop it, the entire certification process will be interrupted. The paper designed a mechanism which uses the already existing chain tail key to certify the chain head of the new key chain.

When the base station node find that key in the one-way key chain has only K_P and K_{P-1}, it will generate a new key chain. First of all the base station will use one-way key chain generation algorithm described above to generate new key chain $L' = \{K_0', K_1', K_2', ..., K_p'\}$. Then the base station sends K_0' to sensor nodes. The node updates its own key information after confirming the signature packet from base station.

The base station node generates a new key chain and broadcast key update message to others.

$$Msg_1 = \{H_d \parallel Cmd_3 \parallel Rnd_i \parallel Ctr \parallel K' \parallel Mac_1\} \quad ..... \quad (10)$$

$$Mac_1 = Mac\{K_{p-1}, H_d \parallel Cmd_3 \parallel Rnd_i \parallel Ctr \parallel K'\} \quad ..... \quad (11)$$

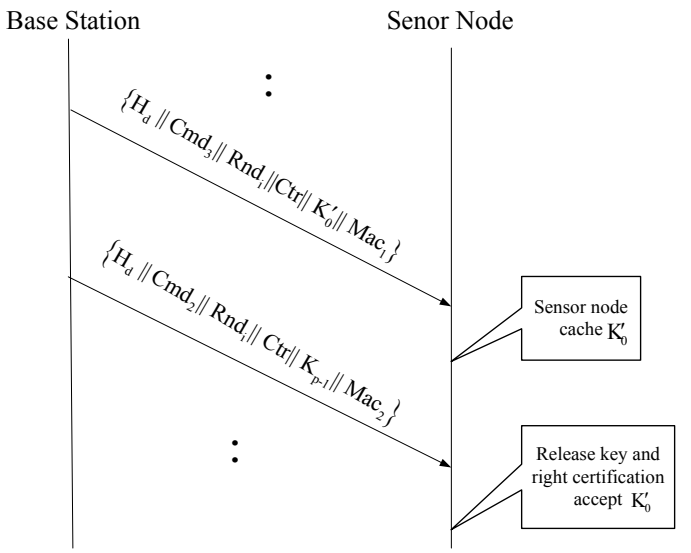

Figure 4. Key chain update

K_0' is the first key of the new key chain, the message use K_{p-1} to carry out signature. Sensor nodes use the Cmd to judge that if this news is the key chain

update message. If the news is, sensor node will cache message and wait for the base station broadcast to release key packets. Sensor node receives a key release message Msg_2.

$$Msg_2 = \{H_d \| Cmd_2 \| Rnd_i \| Ctr \| K_{p\text{-}1} \| Mac_2\} \quad \text{..... (12)}$$

$$Mac_2 = Mac\{K_p, H_d \| Cmd_2 \| Rnd_i \| Ctr \| K_{p\text{-}1}\} \quad \text{..... (13)}$$

By comparing $F(K_{p-1})$ with the last authentication key K_{P-2}, the node judges whether K_{p-1} is a legitimate key. If it is, the node can certify Msg_1. $K_0^{'}$ is the new key when Msg_1 is legal.

c.ii New node authentication

When a new node requests to join in WSN, it needs to register to the base station node and declare its own existence, and obtain the relevant parameters information of the sensor network, including the authentication key.

First the new node must have preset the one-way hash function h=F() and public key information K_{start}.

Step 1: The node first calculates the encryption key and authentication key, the process is as formula 4 and formula 5;

Step 2: To join in WSN, the node sends key request information to the base station. The packet message is as follow:

$$mes\,\mathrm{Re}q = \{\mathrm{Re}q \| ID_{new} \| N_A \| Ctr \| MAC(K_{au}, \mathrm{Re}q \| ID_A \| N_A)\} \quad \text{..... (14)}$$

Where Req represents data's type, and ID_{new} is the node's ID information, N_A is used to achieve the strong refresh authentication.

Step 3: The node monitors the base station that whether it sends the Req response packets. If not, the node will wait for 2s to re-send the key request package msgReq.

Step 4: If the node receives the response packets from the base station：

$$message = \{(K_i \| ID_{new})K_{en} \| Ctr \| MAC(K_{au}, N_A \| (K_i \| ID_{new})K_{en})\} \quad \text{..... (15)}$$

The node will carry out broadcast authentication to the received data packets.

If authentication fails, drop the package. The node waits for 2s to re-send msgReq message to the base station node to request the key information.

Step5: If the authentication is successful, the new node will register at the base station successfully. The node needs to save the key K_i in the response packet.

IV. Simulation

In this paper, Matlab7.0 is adopted to conduct simulations and compare the performance of μTESLA with the proposed ESKA. The simulation results show that ESKA is of lower authentication delay and higher authentication rate in lager broadcast data traffic transmission, and overheads in those applications are reduced.

a. Simulation parameters

It sets the same stimulation scene to μTESLA and ESKA. The stimulation parameters are as following Table 2.

Table 2: Simulation parameters

Form	Explanation
(0,0) to (100,100)	nodes random distribution area
(xBS,yBS)=(50,75)	Coordinates of the base station BS
nn=101	Number of nodes
Emax=2.5J	Node's initial energy
Data Packet's size	500Bytes
Eelec=50nJ/bit	Energy consumption for wireless communication
Stop=360s	Maximum simulation run time

The distributed nodes are shown as in Figure 5.

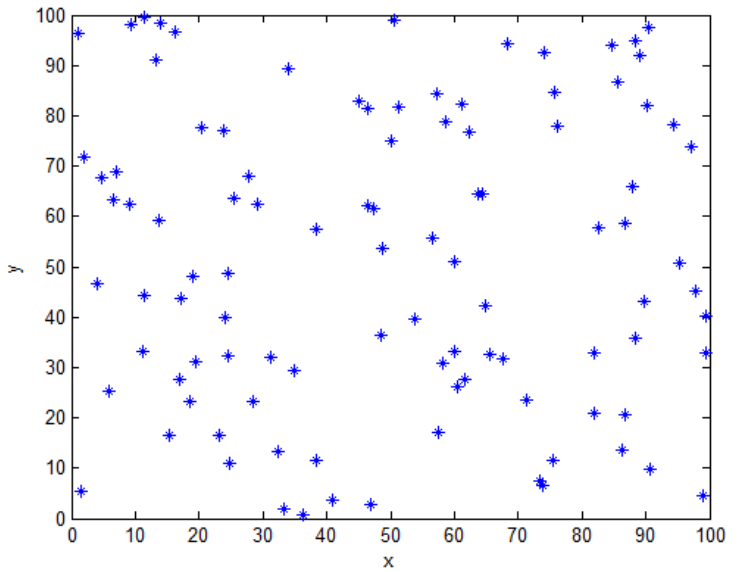

Figure 5. Nodes distribution

b. Simulation analysis

The stimulation experiment compares μTESLA and ESKA in the following aspects: power consumption, authentication rate, authentication delay and packet loss rate.

b.i Power consumption

Figure 6 is energy consumption of the ESKA and the μTESLA with 100s simulation time and key release cycle T = 1s. ESKA consumes less power than μTESLA in the entire simulation cycle. With the base station signature data traffic increasing, the gap between the ESKA and μTESLA on the energy consumption is increasingly smaller. The broadcast authentication scheme ESKA release key packet depends not only on time. When the signature packet traffic is small, the base station will first check whether it sends signature package in last key cycle. If yes, the base station will broadcast key packet; if not, key package will be postponed release until the next key sending cycle, which saves energy by reducing unnecessary keys release.

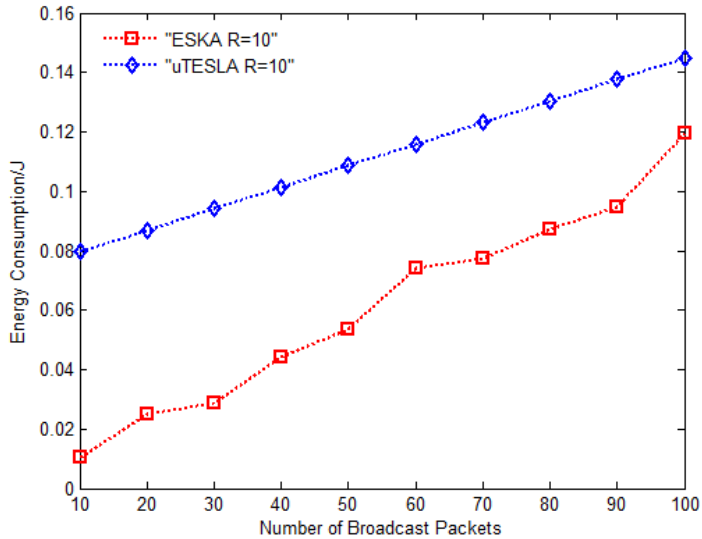

Figure 6. Energy consumption of ESKA and μTESLA

Authentication Rate

The certification rate is the ratio of the sensor node successfully authenticating signature packets and the base station sending signature packets. Figure 7 shows the authentication rate of ESKA and μTESLA in 250s and T=2s.

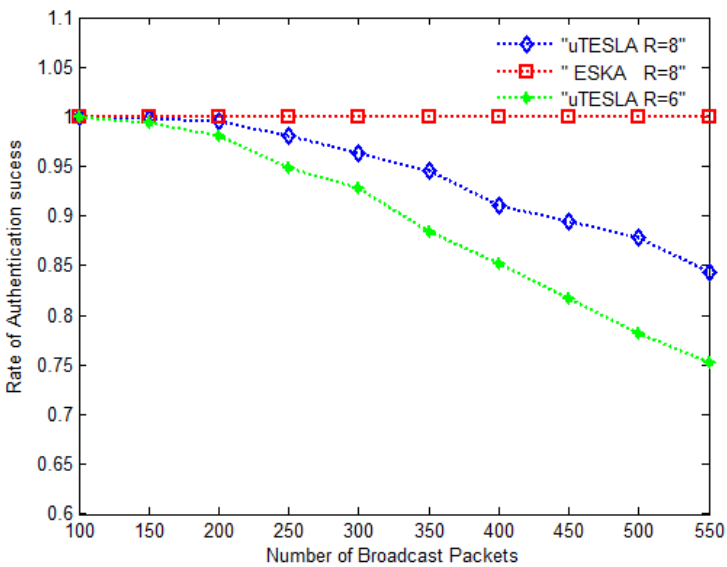

Figure 7. Authentication rate of ESKA and μTESLA

When the threshold value R=8, the ESKA is able to fully certified the signature data packets of base station broadcasted, while the certification rate of μTESLA is decreasing with broadcast traffic of base station increasing.

When R=6, the μTESLA curve declines faster than R=8. Broadcasting the same signature packet of base station, certification rate is low when R=6. In sensor networks with less storage capacity, the network authentication rate of μTESLA will be lower with signature packets' increasing of base station broadcasting.

b.ii Authentication delay

The authentication delay is the length of time from the packet arriving at sensor node to be successfully certificated. Figure 8 is the authentication delay of ESKA and μTESLA.

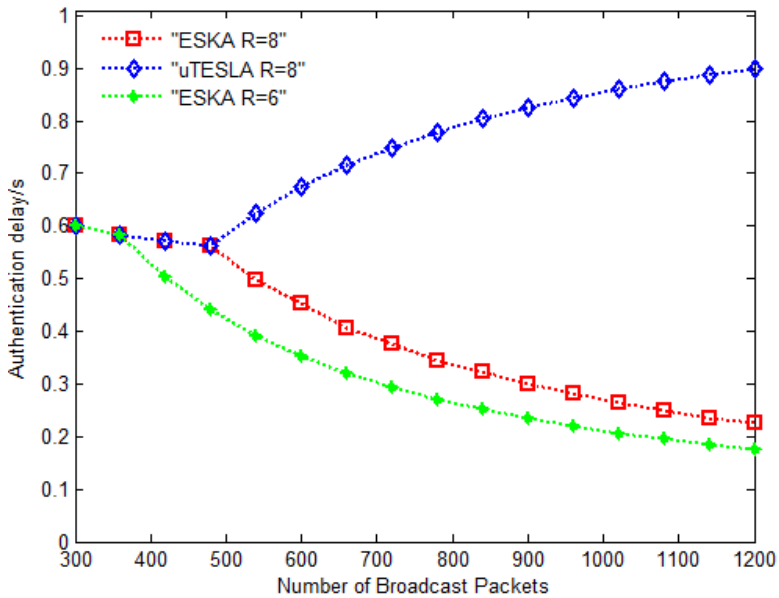

Figure 8. the authentication delay of ESKA and μTESLA with 60s and T=1s

Sensor node can only cache up to 8 signature packets in one network operating cycle. The node can only cache up to 480 signature packages in 60s. When the broadcast traffic exceeds this value, the sensor node using μTESLA protocol will cause the loss of broadcast signature packets because of insufficient cache space. The signature packets received will be certified until key cycle T arrives. When base station broadcasting signature packets is over 480, ESKA will release key packets before key cycle T arriving. With broadcast traffic increasing, the base station will use broadcast signature pack's threshold R as

the condition to release key, so the certification delay will decrease with the increase of traffic. When R=6, the ESKA will release key packets before the signature packets' number is more than 360, the certification delay will decrease with the increase of traffic and is less than R=8.

b.iii Packet Rate

Packet loss rate is defined as the ratio failed signature packages to the signature packets base station broadcasting and sending. The figure 9 shows packet loss rates of ESKA and μTESLA in 150s and T=2s.

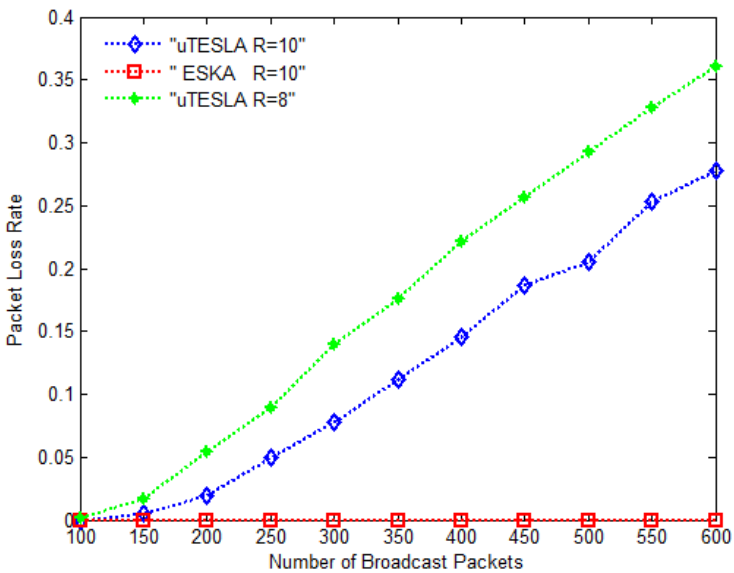

Figure 9. Packet loss rates of ESKA and μTESLA

When threshold R=10, the packet loss rate of ESKA is zero, which because ESKA takes sensor nodes' buffer capacity of signature packets into account. The base station will broadcast immediately authentication key packets when data packets sent over the threshold R in a key release cycle. Sensor nodes can certify these signature packages and release the buffer space. But μTESLA does not take the matter that the traffic will affect the sensor nodes caching the signature packets into account. Packet loss rate will always increase along with the increase of network traffic. When the threshold R=8, the packet loss

rate of μTESLA is higher than R=6, which means that μTESLA has serious loss packet rate in smaller network with limit buffer capacity.

V. CONCLUSION

As one of the most principal security services for wireless sensor networks, broadcast authentication was discussed in this paper. Broadcast authentication is of great significance in promoting the popularity of WSN applications. To satisfy demands from wireless sensor network for broadcast authentication, this paper presented a novel broadcast authentication scheme based on one-way key chain, namely the ESKA. The presented ESKA is basically an asymmetric authentication method which uses the one-way key chain and the delay in releasing the key to perform asymmetric authentication. The algorithm was presented in detail, the authentication rate was increased by adjusting the key release packet interval dynamically according to the broadcast traffic in the network. Meanwhile, the network is prevented from releasing the key packet in no broadcast packet and thus nodes' energy consumption is reduced. A group of simulations are conducted and the results of the presented method are compared to existing method such as μTESLA. The results had demonstrated that the proposed ESKA requires less communication and computation, and is capable to resist the replay attack and DoS attack, without need for time synchronization.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (Grant No.51134023/E0422). We are grateful to express our thanks.

REFERENCES

[1] Vasanth Iyer, Garimella Ram Murthy, M.B. Srinivas, "Training Data Compression Algorithms and Reliability in Large Wireless Sensor Networks[J]", International Journal On Smart Sensing And Intelligent Systems, Vol. 1, No. 4, December 2008, pp. 912-921,.

[2] Wenyan Fu, Deshi Li, Jian chen, Yanyan Han, Jugen Nie, "Topology Optimization Control with Balanced Energy and Load in Underwater Acoustic Sensor Networks[J]", International Journal On Smart Sensing And Intelligent Systems Vol. 4, No. 1, March 2011, pp. 138-159.

[3] Sye Loong Keoh, Emil Lupu, Morris Sloman, "Securing Body Sensor Networks: Sensor Association and Key Management[C]", IEEE International Conference on Pervasive Computing and Communications, IEEE Computer Society, 2009, pp. 53-65.

[4] Du Xiaoming, Chen Yan, "Review on Research status and application of wireless sensor networks [J]", Beijing Business University (Natural Science Edition), Vol. 26, No.1, 2008, pp. 41–44.

[5] Lang Weimin, Yang Zongkai, "Research on wireless sensor network security [J]", Computer Science, Vol. 32, No.2, 2008, pp. 54-58.

[6] James Newsome, Elaine Shi, Song Dawn, "The Sybil Attack in Sensor Networks Analysis& Defenses[C]", In: Proc of Third Intl .Symposium on Information Processing in Sensor Networks (IPSN' 04), Berkeley, California. USA: ACM press, 2004, pp. 259-268, .

[7] Callaway E H. "Wireless Sensor Networks: Architectures and Protocols [M]", Routledge USA: Auerbach Publications, 2004, pp. 45-57 .

[8] Chris Karlof, David Wagner, "Ad hoc Secure routing in wireless sensor networks: attacks and countermeasures [J]",Ad Hoc Networks, Vol. 1, No.2-3, 2003, pp. 293-315,.

[9] T.Jayakumar, C.Babu Rao, John Philip, "Sensors For Monitoring Components, Systems And Processes[J]", International Journal On Smart Sensing And Intelligent Systems, Vol. 3, No. 1, March 2010, pp. 61-74.

[10] Tien-Wen Sung, Ting-Ting Wu, Chu-Sing Yang, Yueh-Min Huang, "Reliable Data Broadcast for Zigbee Wireless Sensor Networks[J]", International Journal On Smart Sensing And Intelligent Systems, Vol. 3, No. 3, September 2010, pp. 504-520.

[11] Mel Siegel, "Scaling Issues in Large Networks of Small Sensors: Energy and Communication Management [J]", International Journal on Smart Sensing and Intelligent Systems, Vol. 1, No. 1, March 2008, pp. 285-299.

[12]Liu Dongang, Peng Ning, "Multi-level μTESLA: Broadcast Authentication for Distributed Sensor Networks[J]". ACM Transactions in Embedded Computing Systems, Vol. 3, No.4, November 2004, pp. 800-836.

[13] Sencun Zhu , Sanjeev Setia, Sushil Jajodia , "LEAP+: efficient security mechanisms for large-scale distributed sensor networks[C]", ACM Transactions on Sensor Networks ,Volume 2 Issue 4, November 2006, pp. 500-528 .

[14] Liu An, Peng Ning, "TinyECC: a configurable library for elliptic curve cryptography in wireless sensor networks[C]", 2008. IPSN '08. International Conference on Information Processing in Sensor Networks, 2008, pp. 245 - 256,.

[15] Arvinderpal S. Wander , Nils Gura , Hans Eberle , "Energy Analysis of Public--key Cryptography on Small Wireless Devices[C]", In: Proceedings of the 3rd IEEE Intl Conference on Pervasive Computing and Communications. California: IEEE Computer Society Press, 2005, pp. 324-328.

[16] Piotrowski Krzysztof, Langendoerfer Peter, Peter Steffen, "How public key cryptography influences wireless sensor node lifetime[C]", Proceeding of the 4th ACM Workshop on Security of AdHoc and Sensor Networks, New York: ACM Press(SASN 2007) , pp. 169-176, .

[17] Ian. F. Akyildiz, Weilian Su, Yogesh.Sankarasubramaniam, Erdal Cayirci. "Wireless Sensor Networks: A Survey [J]", Computer Networks, Vol. 38, 2002, pp. 393-422.

[18] Adrian Perrig,Robert Szewczyk,J.D. Tygar,Victor Wen,David E. Culler, "SPINS: Security Protocols for Sensor Networks[J]", Wireless Networks, Vol. 8, No.5, 2002, pp. 521-534.

[19] Hu.Yih-Chun, Jakobsson.Markus, Perrig.Adrian, "Efficient constructions for one-way hash chains[C]", Applied Cryptography and Network Security. Third International Conference, ACNS 2005. Proceedings (Lecture Notes in Computer Science Vol. 3531), 2005, pp. 423-41.

[20] Kui Ren, Shucheng Yu, Wenjing Lou, Yanchao Zhang, "Multi-user Broadcast Authentication in Wireless Sensor Networks", 2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2007, pp. 223-32.