Reliable Enhanced Secure Code Dissemination with Rateless Erasure Codes in WSNs

Yong Zeng[1]
[1]School of Computer Science and Technology, Xidian University, Xi'an, China
yzeng@mail.xidian.edu.cn

Xin Wang[1,2], Zhihong Liu[1], Jianfeng Ma[1] and Lihua Dong[3]
[2]Education Technology Center, Changchun Institute of Engineering Technology, ChangChun,China
[3]School of Telecommunication Engineering, Xidian University, Xi'an, China
{wangxin, lih_dong, jfma, zhliu}@mail.xidian.edu.cn

***Abstract*—Code dissemination is very useful to remotely fix bugs or add now functions in wireless sensor networks (WSNs) after sensors deployed. Hostile environments keep the secure code dissemination a major concern. The Deluge-based protocols are the widely used code disseminations, however, which have to take much energy and memory to deal with the problem caused by out of order delivery of packets in WSNs. Rateless erasure codes based approaches can reduce the overhead, while failed in defeating DoS attacks. This paper proposed a novel code dissemination scheme, which integrates immediately authentication into rateless erasure codes. The analysis shows that proposed scheme can provide code image confidentiality, bogus code image protection, DoS protection and reliable enhanced property.**

***Index Terms*—wireless sensor networks, code dissemination, reliability, security, rateless erasure codes**

I. INTRODUCTION

Wireless sensor networks (WSNs) can provide wonderful sensing and actuation. WSNs are considered ideal candidates for a wide range of applications, such as industry monitoring, data acquisition in hazardous environments, and military operations [1]. It is often necessary to remotely update sensor nodes' configuration after deployment. For example, it has to fix bugs or add new functionalities. It is very hard to update sensors' softwares one by one due to the large-scale and embedded nature of WSNs. An efficient way is to wirelessly disseminate a code update image and remotely manage the code images on sensor nodes. Such process is so called over the air reprogramming or remote code update. There are two significant steps in over the air reprogramming: code dissemination and code implementation. This paper focuses on how to provide secure and reliable code dissemination.

Deluge[2] is the most well-known code dissemination protocol in WSNs, which is a de facto standard in TinyOS, though, other protocols [3-5] have been suggested. In Deluge the code image is divided into pages, the size of which depends on that of RAM. Each page is split up into packets. Generally speaking the size of packets is about equal to that of frame. The packets are propagated in a pipelined fashion.

However, sensors worked on a hazardous environment. The packets are delivered not in the well defined pipelined fashion due to collisions or multiple parallel transmissions of the same content. It often takes much time and energy to process out-of-order received packets. This is so-called out-of-order-delivery problem, which significantly reduced the reliability of WSNs.

Rateless erasure codes based approaches can reduce the overhead caused by out-of-order-delivery problem. Hgedorn [6] and Rossi [7] proposed efficient code image dissemination scheme based on random linear codes and digital Fountain codes, respectively. In their approaches, the sender generates arbitrarily number of encoded packets using rateless erasure code. Any receivers can get the original code image from any subset of encoded packets, the size of which is equal to or slightly larger than the number of source packets. Note that "any subset of encoded packets" means the receiver can recover the image using out-of-order received packets. As a result their protocols can significantly reduce latency, retransmission, and communication overhead cased by out-of-order received packets. Moreover, due to the rateless property, it is possible to adaptively change the code rate according to the local neighbors' requests or link quality.

However, none of above approaches took security into consideration. The security of Rossi's approach [7] is improved by Bohli [8]. In their approach, the integrity and authentication, two security properties, of each page is achieved by using a digital signature and hash chains closely follows Seluge [1] and [9], which are security extensions from Deluge. However, this approach cannot immediate authenticate each received packets, hence may suffer from DoS attacks by authentication delays of bogus encoded packets. The authors gave a possible

This work is supported by Major national S&T program (No. 2011ZX03005-002), National Natural Science Foundation of China (No.61100235, No.61173135) and the Fundamental Research Funds for the Central Universities.

doi:10.4304/jsw.9.1.190-194

improvement by filtering bogus packets, however, they did not provide detailed effectiveness discussion. LR-Seluge gave another solution by using fixed-rate erasure code and attentively creating hash chains between original and encoded packets using lightweight cryptographic hash functions [10]. However, only receiving sufficient encoded packets to recover one page can the hash images of the next page be recovered. As a result LR-Seluge does not effectively reduce overhead dissemination delays [11].

The above security and reliability enhanced code dissemination schemes are based on rateless erasure codes. Their basic ideas are to bootstrap the code image authentication using a digital signature and to propagate the security of the signature through the code packets by means of hash chains or Merkle hash tree, which is used in Deluge-based protocols. The structure of chains or tree is to keep the packets verifying in order under out-of-order delivery scenarios. For example, in hash tree based proposals, only after successfully receiving *j*th packet of the (*i*+1)th page and successfully verifying its integrity by comparing a hash value, can the integrity of *j*th packet's of the *i*th page can be verified. And the whole code image will be authenticated by a signature of these hash values. The out-of-order delivery will delay the processes of integrity verifying and authentication. However, the rateless erasure code can avoid the out-of-order problem. As a result they cannot take full advantage of erasure codes, and consequently does not immediately authenticate packets.

Our contribution: To the best of our knowledge, available code dissemination schemes do not take their work to be of the interest in immediately authentication with out-of-order-delivery-tolerant property. This paper extends our result [18], which studies a Fountain code using both by Rossi and Bohli, namely the LT code [12]. Our scheme, a reliable enhanced secure code dissemination protocol with immediately authentication property, is achieved by integrating authenticating into LT encoding.

II. PRELIMINARIES

A. Digital Fountain Codes

The basic principle of digital fountain codes, or LT codes, one of rateless erasure Codes, for data transmission can be described as follows. The original data is separated into k packets. Then the source generates a potential unlimited sequence (generally two sizes larger than original data) of code words as follows.

1) To get a code word C_i, a packet degree d_i is randomly chosen following a given distribution function.

2) The encoded packet is d_i packets choosing uniformly randomly out of the k source packets. The d_i packets are successively XORed to get a code word C_i.

Figure 1 illustrates the encoding procedure of LT codes. The encoding is done for at least $n(n>k)$ encoded packets. The coding vector X_i means that packets are XORed for each code word C_i. For example, X_4=(1, 1,0,…,0,1), X_1=(1, 0,1,0…,0,1). X_i may be computed simultaneously by sender and receiver using a pseudo-random number generator with the same seed. However, it will require a strict synchronization between sender and receiver. The alternative scheme is that X_i is appended to each packet.

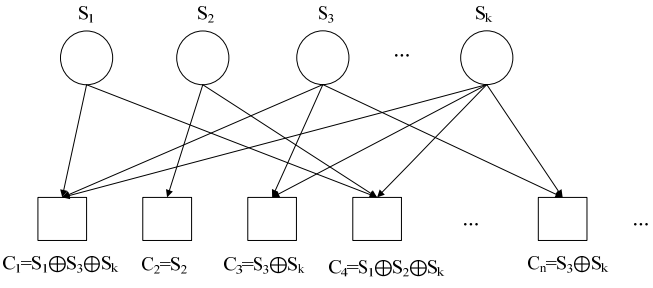

Figure 1. An illustration of LT codes

The receiver extracts the pair (C_i, X_i) from the packet. Then the decoding follows iterative procedure.

1) Find a code word C_i with degree 1. Then it is actually a source packet S_i. If there are none such code words, stop.

2) Find all the other code words containing S_i, then XOR them with S_i, and remove packet i from their coding vectors, i.e., set the coding vector's 'i^{th}' bit 1 as 0 respectively. Goto step 1).

We illustrate the decoding process using the example in Fig.1. Receiver finds that code C_2 is actually source packet S_2. Then set the 2nd bit '1' in X_4 equals to '0' and get new X_4=(1, 0,0,…,0,1) and $C_4=S_1 \oplus S_k$.

The decoding procedure is equivalent to soling a linear equation system $Ax=b$ for x, where $k*k$ matrix A consists of k linear independent coefficient vectors of successfully received codes, and vector b contains the corresponding incoming encoded packets C. The detailed decoding algorithm can be seen in [13].

B. Seluge

The Seluge [1] is considered as one of the most well-known security extension to Deluge. The Figure 2 depicts the Seluge.

Each packet packet $Pkti,j$ in page P_i is augmented to form $p_{i,j}$ by appending the hash value $h(p_{i+1},j)$ of the packet page P_{i+1} (so a hash chain or tree is setup to verify orderly all the packets), where $h()$ is a secure hash function with eight bytes. In Seluge a so-called Merkle hash tree is constructed with M hash values of page P_1. And the page P_0 is created by appending all the authentication hash paths. The root of the Merkle hash tree including some headers is given by a signature packet of code image. Then the packets are disseminated in orders: signature packet first (waiting a few time to make sure that it may arrive majority of all the sensors), then page $P_i(1<i<N+1)$ one by one. If the signature packet and pages arrive in order, then any accepted packets can be immediately authenticated. However, the out-of-order property in WSNs may significantly delay the authentication.

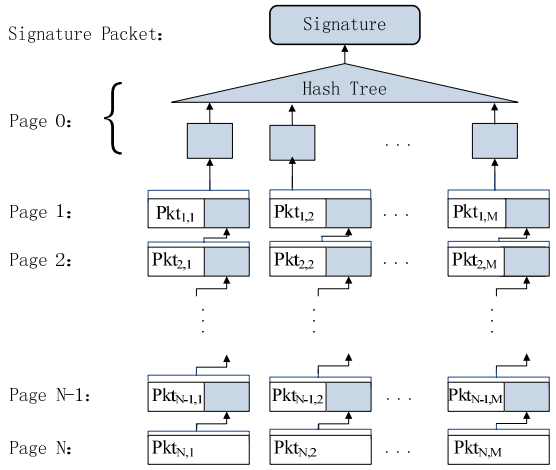

Figure 2. The Seluge code authentication architecture

III. PROPESED CODE DISSEMINATION

We assume that a base station BS is responsible to disseminate update code image on the sensor nodes. It will take multi-hops communication over several sensor nodes to reach all nodes. Sensors are deployed in a untrusted or hostile area.

A. Security Model

We consider the scenario where a code image update takes place in a large scale WSNs with a full and a limited adversary. The limited adversary can eavesdrop or insert packets. The full adversary can eavesdrop, modify and insert packets. The code update process should satisfy the following security requirements:

1) **Code image confidentiality**: the update code image has to be kept secret to prevent eavesdroppers from gaining information for a given time window.

2) **Bogus code image protection**: the unauthenticated update code image should not be written into sensors' memory. This amounts to ensuring authenticity and integrity of the code image.

3) **Denail of Service protoction by immediately authentication**: when an adversary sends modified packets, the honest sensor nodes should not perform unnecessary energy consumption operations. In this paper we focus on the DoS attacks due to the non-immediately authentication problem, which may cause two possible attacks effects: authentication delays or expensive signature verifications.

It is assumed that there is a shared or broadcast key between BS and sensors, which can be distributed by using delayed key disclosure such as μTESLA[14] or pre-distribution[15]. An attacker is supposed keep away from the key.

B. Integrating Authentication into LT Encoding

The basic principle is to integrate authentication code into LT codes so as to immediately verify the integrity and confidentiality of encoded packets arrived in an out-of-order way. More specifically, our proposal differs from existing schemes in that it uses an *authentication code* to verify the integrity. The code is also used to achieve immediately authentication.

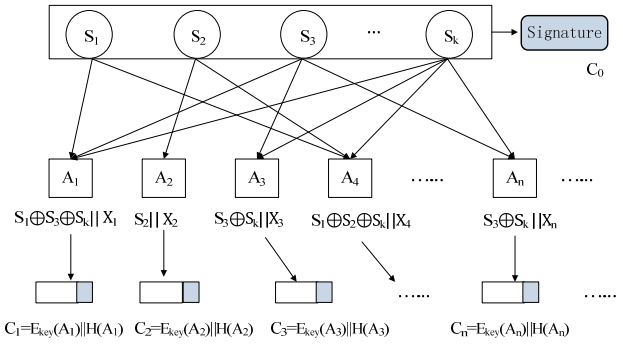

Figure 3. An illustration of LT codes with authentication code

Figure 3 gives an illustration of proposed scheme. In the first step, The update code image is divided into k source packets S_1, …, S_k, then all the k hash values will be signed with a signature scheme to produce signature, i.e., sign($H(S1)||…||H(S_k)$. So a signature packet C_0 is generated, C_0= sign($H(S_1)||…||H(S_k)$). Then they are encoded to A_1, …, A_n using LT code.

Then an *authentication code* C_j of each code is calculated as in equation (1)

$$C_j = E_{key}(A_j)||H(A_j) \quad (1)$$

where $E_{key}()$ is a lightweight encryption algorithm and $H()$ is a secure hash function. In literature conflicting results about encryption algorithm for WSNs have been obtained considering memory requirements, performance or energy consumption [16]. In the code update scenario, the majority amount of ROM is already taken by the reprogramming protocol, so the encryption algorithm should take less ROM memory and execution time. So as to we refer to the hardware ASE-128 block cipher provided by the CC2420 RF chip in the TelosB platform. Then C_j is created by appending plain packet's hash value $H(A_j)$ to encrypted packet $E_{key}(A_j)$. We call C_j an *authentication code*. The confidentiality and integrity of packets can be verified by the authentication code. The detailed security analysis will be shown in Section IV.

C. Transmission , Authentication and Decoding

The signature packet C_0 will be transmitted with Deluge, in which no LT codes are applied. After the signature packet C_0 have been successfully received, the C_1, …, C_k, … will be encoded using the method above. Then the base station will send $n>k$ packets C_i.

When a node received a code, say C_j, then it calculates $D_{key}(E_{key}(A_j))$ to get A'_j, where $D_{key}()$ is the decryption algorithm. If $H(A_j)= H(A'_j)$, then it accepts the packet A_j, it drop the code as a bogus code.

If a node accepts enough corrected codes, say, A'_1, A'_n, then it stops listening and decodes these codes using the nounce and LT decoding algorithm to recover code update image packets, say, S'_1, …, S'_k. Then it calculates C'_0= sign ($H(S'_1)||…||H(S'_k)$). If $C'_0= C_0$, then the code update image is successfully recovered.

If a node has not received enough packets to recover code update image, then it keeps listening. If n packets

have been transmitted, nodes still have not successfully recover code update image, then they send a *NACK* and BS or their neighbors continue sending another n' encoded packets.

IV. ANYLYSIS

This section analyses the features of proposed secure codes dissemination. The LT encoder and decoder are the same with that in [7,8]. So we focus on the security and overhead analysis.

A. Code Image Confidentiality Protection

The plain code is encoded by LT method, and then encrypted using AES-128 block cipher with the shared key. To the best of our knowledge, the best method to break the security of AES-128 without key is the exhaustive search, i.e., brute force attack. In our scheme the attacker is supposed that he or she do not know the key. Thus, the AES-128 encryption of the CC2420 RF chip can provide enough code image confidentiality protection.

This security level is achieved under the consumption that an attacker is keep from the key. However, if an attacker physically captures a sensor node, he or she could compromise the key from its memory. Then the data may be decrypted using the key. However, the compromising key problem can be overcome through the key distribution and updating schemes, which is beyond the scope of this paper. The compromising key does not impair the bogus code image protection and immediate authentication which are our main security goals.

B. Bogus Code Image Protection

The code image is protected by authentication codes. An authentication code C_j, say, $C_j = E_{key}(A_j)||H(A_j)$, is a cascade of the encrypted and hash values of A_j. When a sensor node received C_j, then it calculates $D_{key}(E_{key}(A_j))$ to get A'_j. If $H(A_j)= H(A'_j)$, then it accepts the packet A_j, it drop the code as a bogus code. As a result the code image is secure when the key is secret.

The code image is also protected by the signature packet C_0 transmitted in the first step. The private key to generate the signature is only known to the trusted base station which is responsible for the code dissemination. An adversary cannot get the private key to generate a correct signature. As a result the bogus code image can also be found in the step of signature verification.

Due to the limited memory of sensor nodes, the signature algorithm should take less memory and execution time. The efficient short-lived Rabin-Williams signature scheme [17] is adopted in our scheme.

C. DoS Protection by immediately authentication

The proposed scheme is resistant to the DoS attacks shown in Section IIII from external attackers.

Due to the Authenticate-code-by-Authenticate-code dissemination strategy, upon receiving a code, each sensor node can verify whether the code is a corrected code or not simply by a decryption and a hash operations. Thus, it can immediately authenticate any code it receives, and successfully defeat DoS attacks exploiting authentication delays.

Due to the use of efficient short-lived Rabin-Williams signature scheme, each node can performing a single modular squaring (comparable to a singe hash for RSA-512) and a simple decoding requiring 3-4 hash operations [17] to detect fake signature packets. Thus our scheme provides resistance to DoS attack exploiting expensive signature verifications.

D. Out-of-Order-Delivery-Tolerant

Some works may take much time and memory to process out-of-order received packets. Proposed scheme integrated authentication into LT encoding. Upon detect a correct code from a received packet, each sensor node can simply keep listening until receiving enough packets to recover code update image, where these packets do not need keep order. Thus, proposed scheme is out-of-order-delivery-tolerant.

E. Security Comparison with Previous Approaches

The available code dissemination schemes as Deluge or Deluge-based way do not fully defeat DoS attacks exploiting authentication delays. The reason follows. The Deluge-base schemes need a tree-like structure to keep the packets verifying in order under out-of-order delivery scenarios. Figure 2 has shown the architecture of Seluge(one of Deluge-based schemes). It shows that only after successfully receiving jth packet of the (i+1)th page and successfully verifying its integrity by comparing a hash value, can the integrity of jth packet's of the ith page can be verified. And the whole code image will be authenticated by a signature of these hash values. The out-of-order delivery will delay the processes of integrity verifying and authentication. As a result it cannot fully defeat DoS attacks exploiting authentication delays.

TABLE I.
COMPARISON WITH PREVIOUS APPROACHES

	Code image confidentiality	***Bogus code image protection***	***DoS Protection***		***Out-of-Order-Delivery-Tolerant***
			delays	***verify***	
Seluge[1]	N	Y	N	Y	N
[17]	N	Y	N	Y	N
R-deluge[6]	N	Y	N	N	Y
SYNAPSE++[7]	N	Y	N	Y	Y
[8]	N	Y	N	Y	Y
Our Scheme	Y	Y	Y	Y	Y

Delay: authentication delays; verify: expensive signature verifications

The Fountain code based schemes in [6-8] have the out-of-order-delivery-tolerant property. However, the signature verification of code image is completed after that they receive enough encoded packets and decode them. Since the encoded packets are delivered in plaintext without authentication, an adversary can easily forge fake packets and send to WSNs. If nodes receive fake packets, they know the truth after all the packets are decoded. As a

result they do not defeat DoS attacks exploiting authentication delays.

Table I gives the security comparison with previous approaches. It shows that our scheme not only has the out-of-order-delivery-tolerant property, but also has better security than available.

F. Data Overhead Comparison

The overhead of our scheme should compare with that in [6-8] with out-of-order-delivery-tolerant under the same hash and signature functions.

The communication overhead of our scheme is smaller than those in [6-8]. The reason follows. The sizes of first signature packet C_0 in four schemes are equal if the schemes [6-8] use the same short-lived Rabin-Williams signature efficient as us. Each packet holds one hash value in all the schemes. However, the schemes [6-8] need an additional hash value in the last packet of each page. Let P the number of pages. Then they need transmit more P hash values than us.

The computation overhead of our scheme is smaller than those in [6-8] under DoS attacks. When there are DoS attacks, node must perform more computation in all the schemes. The actual computation depends on the number of fake packets injected by the attacks. Our scheme can immediately judge whether a packets is a fake one or not. However, they [6-8] know after that enough packets are decoded and the signature is computed. As a result, our scheme is the better one under DoS attacks. However, our scheme needs an additional AES-128 operation in each packet, which is to keep the confidentiality of a code image.

V. CONCLUSION

This paper proposed an efficient secure code dissemination scheme with out-of-order-delivery-tolerant property. Proposed scheme can protect code image confidentiality, code image integrity, code image authentication, and defeat external DoS attacks. It has better data overhead than available schemes. The experiment comparison is in hand targeted at the current sensor platforms MicaZ and Imote2. Our scheme can provide security under external attacks. In the future we will discuss the inside attack scenarios.

ACKNOWLEDGMENT

This work was supported in part by a grant from Major national S&T program (No. 2011ZX03005-002), National Natural Science Foundation of China (No.61100235, No.61173135) and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] S. Hyun, P.Ning, A. Liu, and W.Du. “Seluge: Secure and dosresistant code dissemination in wireless sensor networks. In Information Processing in Sensor Networks,” IPSN 2008, pp.445–456. *IEEE*, 2008.

[2] J. W. Hui and D. Culler, “The dynamic behavior of a data dissemination protocol for network programming at scale,” In *Embedded networked sensor systems*, *SenSys ’04*, pp.81–94. ACM, 2004.

[3] D. Estrin, T. Stathopoulos, and J. Heidemann, “A remote code update mechanism for wireless sensor networks,” *Technical Report 30, Center for Embedded Networked Sensing*, UCLA, November 2003.

[4] S.S. Kulkarni and L.M. Wang, “MNP: Multihop network reprogramming service for sensor networks,” In ICDCS ’05: *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, pp.7–16, Washington, DC, USA, 2005.

[5] N. Reijers and K. Langendoen, “Efficient code distribution in wireless sensor networks,” In *WSNA ’03: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pp.60–67, New York, NY, USA, 2003. ACM.

[6] A.Hagedorn, D. Starobinski, and A. Trachtenberg, “Rateless deluge: Over-the-air programming of wireless sensor networks using random linear codes,” In *IPSN ’08: Proceedings of the 7th international conference on Information processing in sensor networks*, pp.457–466, Washington, DC, USA, 2008. IEEE Computer Society.

[7] M. Rossi, N.Bui, G. Zanca, L. Stabellini, R. Crepaldi, and M. Zorzi, “SYNAPSE++: Code dissemination in Wireless Sensor Networks using Fountain Codes,” *IEEE Trans. On Mobile Computing*, Vol.9, No.12, pp.1749-1765, 2010.

[8] J.M. Bohli, A. Hessler, O. Ugus, and D. Westhoff, “Security enhangced multi-hop over the air reprogramming with fountain codes,” in *SenseApp 2009*, Zurich, Switzerland, pp.850-857, October 2009.

[9] O.Ugus, D. Westhoff, and J.M. Bohli, “A ROM-friendly Secure Code Update mechanism for WSNs using a stateful-verifier T-time Signature Scheme,” In *ACM Conference on Wireless Network Security*, *WiSec’09*, pp.29–40. ACM, 2009.

[10] R. Zhang and Y.C. Zhang, “LR-Seluge: Loss-resilient and secure code dissemination in wireless sensor networks,” In *Proceedings of IEEE ICDCS*, 2011.

[11] H. Sangwon, “Secure and reliable code dissemination for wireless sensor networks”, *Ph.D. thesis*, *Raleigh*, North Carolina, 2011.

[12] M. Luby, “LT Codes,” In *Foundations of Computer Science*, *FOCS 2002*, pp.271–282. IEEE, 2002.

[13] M. Mitzenmacher, “Digital fountains: a survey and look forward,” in *IEEE ITW’04*, *San Antonio, TX*, Oct. 2004.

[14] Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler. “SPINS: security protocols for sensor networks”, *Wireless Networks*, 8(5):521–534, 2002.

[15] K. J. Lu, Y. Qian, M. Guizani, and H.H Chen, “A framework for a distributed key management scheme in heterogeneous wireless sensor networks”, *IEEE Transactions on Wireless Communications.2008*, 7(2):639-647

[16] Y.W. Law, J. Doumen, and P. Hartel, “Survey and benchmark of block ciphers for wireless sensor networks”, *ACM Trans. on Sensor Networks*, Vol.2, No.1, pp.65-93, Feb, 2006.

[17] Chae Hoon Lim, “Secure Code Dissemination and Remote Image Management Using Short-Lived Signatures in WSNs”, *IEEE Communication Letters*, Vol.15, No.4, pp.362-364, 2011.

[18] Yong Zeng, Xin Wang, Lihua Dong, Jianfeng Ma, and Zhihong Liu, “Out-of-Order-Delivery-Tolerant Secure Code Dissemination with Fountain Codes in Wireless Sensor Networks”, in CIS’2012: *Proceedings of the 2012 Eighth International Conference on Computer Intelligence and Security*, pp.683-686, Guangzhou, China, November, 2012.