A Lightweight Dynamic Multicast Authentication Scheme

Xuanxia Yao, Xianwei Zhou
School of Computer and Communication Engineering
University of Science and Technology Beijing (USTB)
Beijing, China 100083
kathy.yao@163.com, xwzhouli@sina.com

Xiaojiang Du
Department of Computer and Information Sciences
Temple University
Philadelphia, PA, USA, 19122
dxj@ieee.org

Abstract—**Multicast is a very common communication mode in wireless networks. A security mechanism for multicast is not only the measure to ensure secure communications but also the precondition for other security services. Based on the analysis of Nyberg's fast one-way accumulator and its security, we discover that it has property of absorbency besides the one-way and quasi-communicative properties, which makes it very suitable for applications with variable accumulated items. In this paper, a lightweight dynamic multicast authentication algorithm for small-scale group-based applications is constructed by improving the original Nyberg's fast one-way accumulator. In addition, the security of the algorithm is analyzed in detail and the performance is evaluated in four aspects.**

Keywords-multicast authentication; one-way accumulator; absorbency; quasi-commutative

I. INTRODUCTION

Multicast is a common communication mode in wireless networks. As a fundamental communication approach for many applications and communications based on groups, it can be easily attacked if no security mechanism is used, which can cause further system security problems. Multicast authentication can defend the attacks on multicast. A secure multicast authentication should meet three security requirements, which are verifiability, integrity and non-repudiation. Verifiability means that the data receiver can authenticate the identity of the data originator. Integrity is the property that the data received is not modified. Non-reputation means that the data originator can't deny the data sent by it. It can be said that multicast authentication is not only the crucial measure to ensure the authenticity and integrity of the received multicast data, but also is the premise of many other security services.

With the rapid development of wireless communications and mobile computing techniques, more and more mobile devices, such as cell phones, tablets, and smart sensors begin to form networks, and multicast becomes more common than before. Since these mobile nodes are usually resource-constraint and require lightweight protocols and techniques, it is a very important task to study lightweight multicast authentication algorithms.

In practice, a node often needs to send data to some nodes temporarily, and these scheduled recipients may not belong to a group or only a few of them are members of a group. That is to say, the scheduled recipients of a multicast are variable, which further requires the multicast authentication algorithm to meet the dynamical requirement.

At present, although many multicast authentication mechanisms have been proposed for different applications, to the best of our knowledge, none of the existing mechanisms can meet all the three security requirements and have both lightweight and dynamic properties.

In this paper, a lightweight dynamic multicast authentication scheme based on Nyberg's fast one-way accumulator [1] is proposed for small scale applications. Using the proposed scheme, multicast authentication can be realized easily as long as the data sending node has a shared key with each of the scheduled recipients.

The remainder of this paper is organized as follows: In Section II, we review the related work on multicast authentication. In Section III, we give an introduction for the one-way accumulator and the Nyberg's fast one-way accumulator. In Section IV, we present our improved Nyberg's fast one-way accumulator for multicast authentication. In Section V, we describe the process of dynamic multicast authentication, and present detailed security analyses. In Section VI, we evaluate the performance of the proposed authentication mechanism. Section VII concludes this paper.

II. RELATED WORK

Typically, an asymmetric mechanism is required to implement multicast authentication, because the multicast environment is asymmetric, and in most cases the receivers of multicast messages don't trust each other [2, 3, 4].

Current researches on multicast authentication can be classified into 3 categories, which are public-key based multicast authentication [5, 6]; symmetric key based multicast authentication [7, 8] and hybrid multicast authentication [9, 10].

Public-key based cryptographic algorithms have the nature of asymmetry, which make them be very suitable for multicast authentication and have an advantage in cases where the data receivers are uncertain. However, the high computation, communication and storage overheads make them impractical for resource-constraint nodes. Although many improvements

This work is supported by Chinese National Scholarship Fund and Chinese National High Technology Research and Development Program 863 under Grant No. 2012AA121604, as well as the US National Science Foundation (NSF) under grants CNS-0963578, CNS-1022552, and CNS-1065444.

978-1-4799-5970-9 © 2014 IEEE

have been made on them so that they can be used in resource-constraint environment, such as the lightweight public-key infrastructure based on elliptic curve cryptography [11], the heavy overhead is still a drawback.

The symmetric key based multicast authentication is essentially MAC (Message Authentication Code) or hash value based multicast authentication. The MAC generated directly by the group key is not fit for multicast, because the property of non-repudiation can't be realized. The existing MAC based multicast authentication schemes usually employs the shared key between the data sender and the scheduled receiver other than the group key to generate the MAC. The key-ring based multicast authentication [4] is a typical one, whose high communication overheads and poor scalability make it impractical. µTESLA [7] is another MAC based multicast authentication scheme, based on which, there are also some µTESLA-like schemes [8]. They use time to realize asymmetry and can authenticate the multicast source and the integrity of the multicast data by employing a one-way hash chain. Compared with public-key based solutions, they have lower computation overheads, but they have to suffer from serious DoS attacks [12] due to the delayed authentication. In order to counteract these problems, Merkle hash tree based multicast authentication schemes are provided [13, 14], but they introduce high communication costs due to the long signature for reach message. In addition, One-time signature based authentication schemes are also attributed into this kind, which is based on one-way hash chains, whose signature length is too long to be practical [2, 15].

The hybrid multicast authentication schemes refer to those schemes that exploit both public-key cryptosystem and MAC or hash functions. The digital stream signature is a typical one [9], where the digital stream is divided into several packets and a chain of hashes is used to link each packet to the one preceding it. The packet chain can be authenticated by the traditional digital signature on the first packet and hash values of the rest packets. Although the overhead is low, it can't resist packet loss. In order to resist packet loss, some improved scheme have been presented [10], the basic idea is to append the hash of each packet to more than one place in the stream.

It is obvious that hash and MAC functions are usually used to achieve lightweight multicast authentication. In this paper, we design a MAC based multicast authentication in a new way after analyzing and revising Nyberg's fast one-way accumulator.

III. Nyberg's Fast One-Way Accumulator

A. One-Way Accumulator

The concept of one-way accumulator is proposed by Benaloh and Mare [16] for member testing. It is an alternative to digital signatures for credential authentication to verify whether one value is in the specified set or not.

A one-way accumulator is indeed a one-way hash function with the quasi-commutative property. The function f: X × Y→X is said to be quasi-commutative if for all $x \in$ X, and for all $y_1, y_2 \in$ Y.

$$f(f(x,y_1),y_2) = f(f(x,y_2),y_1) \tag{1}$$

Let H be a one-way hash function with quasi-commutative property, if one starts with an initial value $x \in X$ and all $y_1, y_2, ... y_n \in Y$, the accumulated hash value is computed by (2).

$$Z = H(H(\cdots H(H(x,y_1),y_2)\cdots y_{(n-1)}),y_n) \tag{2}$$

In the accumulated hashing, the accumulating items are cumulatively hashed together, and H can ensure that the accumulated hash value doesn't depend on the order where the items appear in the list.

When a one-way accumulator is used for member testing, all members keep the accumulated hash value Z, for member i, z_i is its partial accumulated value of all other members and y_i is its accumulating item. When it is required to prove that it is one of the members, it presents its y_i and z_i, any other members can compute $H(z_i,y_i)$ and verify whether $H(z_i,y_i)$ is equal to Z. If yes, node i is one of the members.

Nowadays, one-way accumulators are usually constructed on public key cryptographic algorithms, such as RSA or ECC, and the costs for computing and memory are very high.

B. Nyberg's Fast One-way Accumulator

Nyberg's fast one-way accumulator is not a trapdoor function, which is different from other one-way accumulators. It is based on the general hash function and simple bitwise operation, and fast accumulating operations can be achieved.

Let $N=2^d$ be the maximum number of the accumulating items, where d is a positive integer. Let $x_1,. . . , x_m$ be the items to be accumulated, here $m \leqslant N$. It is assumed that h is a one-way hash function, which can map bit strings of arbitrary length to bit strings of length l. For each accumulated item x_i, its hashing value $y_i=h(x_i)$, here $i=1,\dots,m$. Let the length of the accumulated hash value be r. The relations among l, r, and d can be expressed by $l = r \times d$.

For each y_i, it is divided into r substrings of length d and denoted as: $y_i=(y_{i,1},...,y_{i,r})$, $y_{i,j}$ is the j^{th} bits string of length d. If $y_{i,j} \neq 0$, it is replaced by 1. If $y_{i,j}$ is a string of d bits zeros, it is replaced by 0. So the y_i of length l can be mapped to a string b_i of length r and can be denoted as: $b_i=\alpha_r(y_i)= (b_{i,1},\dots,b_{i,r})$, here $b_{i,j}$ is the j^{th} bit of b_i. If h is an ideal hash function, b_i can be considered as the value of r independent binary random variables, for which the probability of $b_{i,j}=0$ can be estimated by $\text{p}(b_{i,j}=0) =2^{-d}$.

Let $H^{Nyb}(K_Z,Y)$ be the Nyberg's accumulating function, the accumulated hash value on Y can be estimated by (3).

$$Z(Y) = H^{Nyb}(K_Z,Y) = K_Z \otimes \alpha_r(h(X)) \tag{3}$$

In (3), K_Z is the initial value, ''⊗'' is used to denote logic multiplication. For any y_i and y_j, there should be (4) and (5).

$$H^{Nyb}(H^{Nyb}(K_Z,y_i),y_j) = K_Z \otimes \alpha_r(h(x_i)) \otimes \alpha_r(h(x_j)) \tag{4}$$

$$H^{Nyb}(H^{Nyb}(K_Z, y_j), y_i) = K_Z \otimes \alpha_r(h(x_j)) \otimes \alpha_r(h(x_i)) \quad (5)$$

Because logic multiplication obeys commutative rule, (6) should be true, which indicates that H^{Nyb} (K_Z,Y) has the quasi-commutative property.

$$H^{Nyb}(H^{Nyb}(K_Z, y_i), y_j) = H^{Nyb}(H^{Nyb}(K_Z, y_j), y_i) \quad (6)$$

Since $H^{Nyb}(K_Z,Y)$ is constructed on the one-way hash function h, and also inherits the h's one-way property. The properties of one-way and quasi-commutative property make $H^{Nyb}(K_Z,Y)$ be an accumulator.

For the accumulated items set X={x_1 , … , x_m}, the accumulated value of Nyberg's fast one-way accumulator can be denoted by（7).

$$Z = H^{Nyb}(K_Z, Y) = K_Z \otimes \prod_{i=1}^{m} \alpha_r(y_i) = K_Z \otimes \prod_{i=1}^{m} \alpha_r(h(x_i)) \quad (7)$$

Here, let z_j be the j^{th} bit in Z. In order to verify that x_i is an accumulated item of Z, you should compute y_i=$h(x_i)$ and map y_i to b_i = ($b_{i,1}$ . . . . ,$b_{i,r}$,) ,which is a binary string of length r. For all j = 1,..., r, only if $b_{i,j}$=0, there is z_j =0, which indicates x_i is an accumulated item of Z, otherwise, it indicates x_i is not an accumulated item of Z.

In addition, logic multiplication satisfies absorbency, which makes "$A \otimes A = A$". Equation (8) should be true.

$$H^{Nyb}(H^{Nyb}(K_Z, y_i), y_i) = H^{Nyb}(K_Z, y_i) \quad (8)$$

Consequently, the accumulated value Z can substitute partial accumulated value z_i to be the member i's witness information of Nyberg's fast one-way accumulator. That is to say z_i=Z for any x_i. And $H^{Nyb}(Z,Y)$ can be the verify function for it. In this way, if $H^{Nyb}(Z,y_i)$=Z, then x_i is an accumulated item of Z, otherwise, x_i is not an accumulated item of A.

C. Security Analysis of Nyberg's Fast One-way Accumulator

The security of Nyberg's Fast One-way Accumulator depends on the difficulty to forge an accumulated item successfully, which further depends on the randomness properties of the hash function h. Nyberg has the following Theorem 1 [1] to prove its security.

Theorem 1. Let b_{ij} and c be independent binary random variables such that Pr($b_{i,j}$ =0) = Pr(c_j=0)=2 $^{-d}$, for i = 1,..., m (m≤N=2^{-d})and j = 1,..., r . Let a =(a_1 ,..., a_r) be the coordinate-wise product of the r-tuples b_i = ($b_{i,1}$,...,$b_{i,r}$). Then the probability that, for all j = 1,..., r, we have c_j=0 only if a_j =1, is equal to $2^{-d}(1-2^{-d})^m$.

If we consider c as the result that a forged accumulated item is hashed and mapped by rule α_r, for each j = 1,... ,r, the probability that c_j = 0 and a_j = 1 can be depicted by (9), where e is Neper number, and P_F is also the probability of attacking successfully.

Let t = r/(N×e), there is r=N×e×t. Here, t is called security level.

$$P_F = \left(1-2^{-d}\left(1-2^{-d}\right)^m\right)^r \le \left(1-\frac{1}{N}\left(1-\frac{1}{N}\right)^N\right)^r \approx e^{-\frac{r}{N\times e}} \quad (9)$$

According to (9), when t is big enough, P_F is small enough and the security of Nyberg's fast one-way accumulator can be considered strong enough.

The length of the hash code l =r×d=N×e×t×d. It can be seen that the length of the hash code in Nyberg's fast one-way accumulator depends on t and d. Let the maximum number of accumulated items N be 2^{10} and t be 10, the length of the hash code is 278 Kbits, which is much longer than the hash code (128-512 bits) of existing hash functions. There are several ways to get the required long hash code.

IV. The Improved Nyberg's Fast One-Way Accumulator

In general, a one-way accumulator can be used to mutual-authenticate members of a group comprised of fixed members. Nyberg's fast one-way accumulator has the property of absorbency besides the properties of one-way and quasi-communicative, which makes it not only suitable for member testing but also fit for the applications in which the accumulated items are dynamic. These characteristics inspire us to apply it to dynamic multicast authentication. In [17], we use MACs directly as the accumulated items of the Nyberg's fast one-way accumulator for multicast authentication and only consider sending data to all the neighbors or group members of the sender. In this paper, we improve Nyberg's fast one-way accumulator in two aspects to further simplify the computing process in multicast authentication. At the same time, the original Nyberg's fast one-way accumulator is used to help a receiver identify whether it is a scheduled receiver or not.

Firstly, the accumulation item is changed from a single data to a two-tuples. That is to say that each accumulation item is made up of two elements. One is the shared key between the multicast source and the receiver, the other is the multicast data. The difference between 2 accumulation items is the shared key, which is only known to the sender and the receiver.

Secondly, a HAMC function is used to replace the hash function h in Nyberg's fast one-way accumulator to embed the MAC's computing process into the accumulator. Since a HMAC function has all the properties of a hash function, the replacement can reach the same effects as the original one.

The improved Nyberg's fast one-way accumulator can be described by (10).

$$Z(Y) = H^{Nyb}(K_Z, Y) = K_Z \otimes \alpha_r(HMAC(X)) = K_Z \otimes \prod_{i=1}^{m} \alpha_r(y_i)$$
$$= K_Z \otimes \prod_{i=1}^{m} \alpha_r(HMAC(K_{S,i}, Data)) \quad (10)$$

The security of the improved Nyberg's fast-one way accumulator is similar to the original one, which eventually depends on the randomness properties of the HMAC function. In addition, its security also hinges on the length of the accumulation value r and becomes stronger with the increment of r when the number of accumulation items is fixed.

Based on the improved and the original Nyberg’s fast one-way accumulator, we can conveniently achieve lightweight dynamic multicast authentication.

V. Description of the Scheme

A. Application Model

It is assumed that the multicast authentication scheme is used in the applications with the following two assumptions.

- All nodes in the Network can run the original and improved Nyberg’s fast one-way accumulator.
- A node can establish a shared key with other nodes (such as its neighbor nodes or nodes in its group) using the existing key management scheme.

B. Notations

To describe the proposed mechanism, the notations used are listed as following:

$K_{i,j}$: the shared key between node i and node j.

S: multicast source.

R_i : multicast receiver i.

ID_j: the ID of node j.

m: the number of the scheduled data receivers.

Z: the accumulated value computed of the MACs by the multicast source.

Z': the signature received by a node.

A: the accumulated value of the scheduled receivers IDs, which is computed by the multicast source.

Data: the multicast data.

M: the multicast message, M= “ID_s ||*Data*||*Z*||*A*” .

($K_{S,I}$,*Data*): the accumulated item for the improved Nyberg’s fast one-way accumulator.

C. The Proposed Multicast Authentication Scheme

The proposed multicast authentication scheme is based on the improved and the original Nyberg’s fast one-way accumulator. The improved one is used to embed both the secret information (i.e., the shared keys) and multicast data into the accumulated value, and the accumulated value MACs can be served as the signature of the multicast. If the signature passes the receiver’s verification, it is that the MAC from the shared key and the received data is a part of the accumulated value, which indicates that the multicast data comes from the claimed source and is unchanged. The original one is used to accumulate the IDs of the scheduled receivers so that a receiver can identify whether it is a scheduled receiver or not.

For the sake of simplicity, we assume that node S wants to send data to some of its neighbors to illustrate the multicast authentication process. To generate a multicast message for the multicast data, four steps should be done.

Step 1. Node S uses the keys sharing with each scheduled neighbor i and the multicast data to construct an accumulating items set of ($K_{S,i}$,*Data*) .

Step 2. Node S computes the accumulated value Z on each item in ($K_{S,i}$,*Data*) according to (11). Although the accumulated key K_z can be randomly selected, for the sake of security (the analysis is given in Section VI), there must be enough bits of “1” in the accumulated value, and we make $K_z=\{1\}^r$.

$$Z = Z(Y) = K_Z \otimes \prod_{i=1}^{m} \alpha_r (HMAC(K_{S,i}, Data)) \qquad (11)$$

Step 3. Let $x_i=ID_i$, (i=1,…,m), $K_z=\{1\}^r$. Calculating the accumulated value A of each scheduled receiver’s ID by (7).

Step 4. Multicast message M = “ID_S ||*Data*||*Z*||*A*” is constructed and sent out.

When a neighbor node i receives such a multicast message M, eight steps should be done for authentication.

Step 1. The ID of the claimed multicast source S, *Data*, the received signature Z' (which is correspond to the original value Z) and A should be extracted from M, respectively.

Step 2. Node i checks whether it has a shared key with the multicast source according to ID_S. If not, node i is not a scheduled receiver, it will either forward this message or discard it according to the scheduled policy, and go to step 8.

Step 3. Node i calculates $h_i=h$ (ID_i) and maps h_i according to α_r, which can denoted by $\alpha_r(h_i)$.

Step 4. Node i checks whether the equation $A=A\otimes\alpha_r(h_i)$ is true or not, if it is true, it indicates that node i is one of the scheduled receiver, otherwise it will either forward this message or discard it according to the policy and go to step 8.

Step 5. Node i checks whether the number of “1” bits in signature Z' can meet the requirements (the requirements and the reason are given in subsection D of this section) or not. If not, the multicast message is considered as a forged one and should be discarded, else go to step 8.

Step 6. Node i uses the shared key between S and itself to compute $y_i=HMAC(K_{S,i}, Data)$ and maps y_i to $b_i=\alpha_r(y_i)$.

Step 7. If $Z'= Z'\otimes b_i$, it indicates that this multicast message is from node S and the multicast data is not changed. Otherwise, it indicates that the multicast data has been changed or the multicast message is not from node S.

Step 8. Stop.

D. Security Analysis

For the sake of convenience, the security level t in subsection C of Section Ⅲ is adopted and the event with a probability less than e^{-t} is considered as impossible. A successful attack is defined as any receiver believing that a forged multicast packet is from the claimed sender. If the probability of a successful attack is less than e^{-t}, the mechanism is considered secure at t security level.

Since the accumulated value of IDs of the scheduled receivers is not helpful to land a successful attack, attackers are not interested in it. Here, we only discuss the security of the signature or the accumulated value of ($K_{S,i}$,*Data*), which is the real target of an attacker and is critical to the security of the algorithm.

The signature in the proposed multicast mechanism is constructed on the improved Nyberg's fast-one way accumulator, so its security should be analyzed on the security requirements of the improved accumulator, which means that the length of the accumulated value r should satisfy $r \geq N \times e \times t$ for a fixed N and t.

Compared to the Nyberg's fast one-way accumulator, the signature verification process is a reverse process. A receiver authenticates the authenticity and integrity of the multicast data by checking whether the equation $Z' \otimes b_i = Z'$ is true or not.

Since $K_{S,i}$ is only shared by the claimed sender and the receiver R_i, the attacker could not forge b_i or know b_i in advance. A success attack means that a forged signature can make a receiver believe that it comes from the claimed sender. If there are more "0" in the forged signature, then it is easier for the forged signature to pass the verification. And when $Z'=\{0\}^r$, $Z' \otimes b_i = Z'$ is always true whatever b_i is, which means that an attack can always been successfully launched. This situation should not happen in the multicast authentication mechanism. It is necessary to analyze the distribution of "1" in a normal accumulated value so as to prevent the attacker from forging all "0" bits or having too many "0" bits in a signature.

Let q be the number of "1" in a normal signature. According to the normal accumulating process, the probability of each bit's value in an accumulated value Z meets the binomial distribution. Assume that there are m ($m \leq N$) accumulated items. Let Z_i be the i^{th} bit in Z, we have $P(Z_i=0)=(1-2^{-d})^m$ and $P(Z_i=1)=1-(1-2^{-d})^m$.

Accordingly, the probability that there are q bits "1" in Z can be estimated by (12).

$$P(q) = \binom{r}{q}\left(\left(1-2^{-d}\right)^m\right)^q \times \left(1-\left(1-2^{-d}\right)^m\right)^{(r-q)} \tag{12}$$

It can be seen from (12) that q depends on r, and r further depends on d and the security level t for a given accumulated item set.

Based on the probability theorems, the probability of q>k can be estimated by (13), where k is between 0 and r.

$$\begin{aligned} P(q>k) &= 1-P(q\leq k) \\ &= 1-\sum_{i=0}^{k}\binom{r}{q}\left(\left(1-2^{-d}\right)^m\right)^i \times \left(1-\left(1-2^{-d}\right)^m\right)^{(r-i)} \end{aligned} \tag{13}$$

The minimum q can be computed by (13), which depends on d, m and r. and r further depends on d and t. For a multicast application, d, m and t can be known, so the minimum of q only depends on r.

Since the minimum q can be known in advance for a given application, the number of "1" in the signature can be set to be more than the minimum of q, which makes attacks more difficult, and also helps a receiver distinguish a forged multicast quickly. For instance, let the number of "1" in the signature be q', a multicast with $q'<\min(q)$ is considered as fraudulence and needs not to be further verified. Only a multicast with $q' \geq \min(q)$ needs to be further authenticated.

The receiver R_i will compute HMAC on the received multicast data using the shared key between itself and the claimed sender and map it to get b_i. The probability of $b_{i,j}=0$ can be estimated by $P(b_{i,j}=0)=2^{-d}$, the probability of $b_{i,j}=1$ can be estimated by $P(b_{i,j}=1)=1-2^{-d}$. Assuming that the number "1" in a forged signature is q', the probability of any bit in the forged signature being equal to 1 can be estimated by $p(Z'_j=1)=q'/r$. The probability of cheating a receiver to trust the signature can by estimated by (14).

$$P_F = \left(1-\left(2^{-d}\times\frac{q'}{r}\right)\right)^r = \left(1-\frac{q'}{N\times r}\right)^r \tag{14}$$

Equation (14) shows P_F depends on q', N and r, and q' further hinges on r according to (13). Given N, P_F only depends on r, so we can enlarge r to decrease the probability of attack successfully. Let security level $t=10$, that is P_F should less than e^{-10}(4.53999E-05). We can compute the minimum r by (14), and q' can be calculated by (13).

When N=4, 8, 16, 32, the value of the minimum q, r and P_F are shown in TABLE I. For the sake of simplicity, the value of r is the integer times of one byte.

TABLE I. THE RELATIONS AMONG N, MINIMUM Q, R AND P_F

d	**N**	**Minimum q(bits)**	**Minimum r(bytes)**	**P_F**
2	4	40	26	3.29467E-06
3	8	80	40	1.00566E-05
4	16	160	72	2.1099E-05
5	32	320	132	3.04606E-05

We can see from Table I that the length of signature r increases sharply with the increment of the number of the accumulated items when t is fixed, which indicates that the multicast authentication mechanism based on the improved Nyberg fast one-way accumulator is not fit for the multicast to large number of receivers but very suitable for small scale applications with limited resources.

VI. PERFORMANCE EVALUATION

From Section V, we can see that the proposed scheme can meet the verifiability, integrity and non-repudiation requirements for multicast authentication completely. Here, we put focus on instantaneity, overhead, robustness and dynamics to evaluate the performance of it.

A. Instantaneity

Here, instantaneity has two meanings, one is immediate authentication, and the other is that one can multicast a message at irregular time.

For the former, since one multicast message in the proposed algorithm includes all the information that is required by the process of authentication, a scheduled receiver can authenticate the multicast data immediately as soon as it receives the entire multicast message.

For the latter, the proposed multicast authentication scheme doesn't depend on time factor, and each multicast message is independent, therefore, a multicast data can be sent at any time.

B. Overhead

The overhead mainly includes computation overhead and communication overhead.

The proposed scheme has low computation overhead. In the process of generating and verifying a signature, the main operations can be classified into 3 types: logical multiply, HMAC and bits mapping. They are all simple bit operations in nature and very helpful to improve execution efficiency. Hence, the mechanism is fit for resource-constraint situations.

The communication overhead depends on the length of the multicast message. In the proposed scheme, it is comprised of the ID of the multicast node, data and the signature. Since the first 2 parts are always prerequisite, the communication overhead can be measured only by the length of the signature. The length of our signature is longer than that of μTESLA, and much less than that of key ring based schemes. In small scale applications, the signature length of ours is not very long, which means low communication overhead. However, in large scale applications, the increase of the signature's length will cause high communication overhead.

C. Robustness

Robustness also has two meanings, one is the ability to resist node compromising, and the other is ability to tolerate packet loss.

In the proposed scheme, the signature is generated by the multicast data and each shared key between the multicast source and every scheduled receiver. If one receiver is compromised, it will not affect other receivers to verify the signature and communication. Only the communications between the compromised node and the nodes who share secret key with it become insecure. So we can say our algorithm can resist node compromising.

Our scheme also can tolerate packet loss, because each multicast message is independent of others and an attacker can't refer any secret information form a packet, the robustness to packet loss can be guaranteed.

D. Dynamics

The dynamics refers that the scheduled receivers are variable and can be specified as needed. In our scheme, a node can multicast to any nodes as long as the sender has a shared key with each of the scheduled receiver, it is obvious that the property of dynamics can be achieved.

VII. Conclusions

Multicast is a common communication mode in various types of networks. Lightweight multicast authentication is needed to achieve secure and efficient multicast in resource-constraint environment. By utilizing the absorbency property of the original Nyberg's fast one-way accumulator, we improve the Nyberg's fast one-way accumulator, based on which, we construct a lightweight multicast authentication scheme for small scale applications. Its security properties is analyzed in details. In addition, we evaluate its performance from four aspects. The analysis and evaluation results show that the proposed dynamic lightweight multicast authentication scheme can meet the requirements for multicast authentication [2].

References

[1] K. Nyberg, "Fast accumulated hashing," in Proc of the 3rd Fast Software Encryption Workshop,Berlin: Springer-Verlag, 1996. pp.83–87.M. Luk,

[2] M. Luk, A. Perrig, Br. Whillock, "Seven Cardinal Properties of Sensor Network Broadcast Authentication," in Proc. SASN'06, Alexandria, Virginia, USA, 2006, pp.147–156.

[3] D. Boneh, G. Durfee, and M. Franklin, "Lower bounds for multicast message authentication," in Proc. Advances in Cryptology — EUROCRYPT '01, 2001, pp.434–450.

[4] R.Canneti, J. Garay, G. Itkis and et al. "Multicast security : a taxonomy and some efficient construction," in Proceedings of INFOCOM '99 Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, New York, U.S.,Mar,1999, vol.2, pp.708–716.

[5] X. Cao, W. Kou, L. Dang, B. Zhao, "IMBAS: identity-based multi-user broadcast authentication in wireless sensor networks," Computer Communications, 2008, vol. 31, no.4, pp.659–667.

[6] V. Gupta, M. Millard, S. Fung, and et.al, "Sizzle: A standards-based end-to-end security architecture for the embedded internet," in Proceedings of the Third IEEE International Conference on Pervasive Computing and Communication (PerCom), 2005, pp.247–256.

[7] A. Perrig, R. Szewczyk, V. Wen, D. culler, J.T ygar, "SPINS:security protocols for sensor networks," in Proc. Seventh Annual ACM International Conference on Mobile Computing and Networks(Mobicom 2001),Rome, Italy, July 2001,pp.189–199.

[8] D. Liu, P. Ning, "Multi-level μTESLA: broadcast authentication for distributed sensor networks," ACM Trans on Embedded Computing Systems, 2004, vol.3, no.4, pp.800–836.

[9] R. Gennaro， P. Rohatgi, "How to sign digital streams," Information and Computation，2001，165(1), pp.100–116.

[10] H. K. Aslan, "A hybrid scheme for multicast authentication over lossy network," Computers & Security, 2004, vol. 23, pp. 705–713.

[11] D. Malan, M. Welsh, M. Smith. "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography," in Proceedings of IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON), October 2004, pp.71–80.

[12] P. Ning, A. Liu, W. Du, "Mitigate DOS attacks against broadcast authentication in wireless sensor networks," ACM Trans on Sensor Networks, 2008, vol.4, no.1, pp. 1–35.

[13] C. K. Wong and S. S. Lam, "Digital signatures for flows and multicasts," in Proceedings of the 1998 International Conference on Network Protocols (ICNP'98) [C], Austin , Texas , Oct . 1998, pp. 198–209.

[14] Y. Hu, A. Perrig, D. B. Johnson, "Packet leashes: A defense against wormhole attacks in wireless networks," in Proceedings of IEEE INFOCOM, April 2003, vol.3, pp.1976-1986.

[15] L. Reyzin and N. Reyzin, "Better than BiBa: Short one-time signatures with fast signing and verifying," in Proc. ACISP 2002, Jul. 2002, pp.144–153.

[16] J. Benaloh, M. D. Mare, "One-way accumulators: A decentralized alternative to digital signatures," in Proc. Advances in Cryptology-Eurocrypt'93, LNCS 765. Berlin: Springer-Verlag, 1993, pp.274–285.

[17] X. Yao, X. Zheng, X. Zhou, "Broadcast authentication algorithm for wireless sensor networks," Tongxin Xuebao/Journal on Communications. Nov. 2010, vol.31, no.11, pp.49–55.