A Lightweight Multicast Authentication Mechanism for Small Scale IoT Applications

Xuanxia Yao, Xiaoguang Han, Xiaojiang Du, *Senior Member IEEE,* and Xianwei Zhou

***Abstract*— Security is very important for Internet of Things (IoTs). As a main communication mode, the security mechanism for multicast is not only the measure to ensure secured communications, but also the precondition for other security services. With the analysis of Nyberg's fast one-way accumulator and its security, we discover that it has the property of absorbency in addition to the one-way and quasi-communicative property that makes it very suitable for applications in which accumulated items are dynamic. In this paper, we revise the original Nyberg's fast one-way accumulator and construct a lightweight multicast authentication mechanism for small scale IoT applications. We analyze the security of the mechanism in detail. In addition, we evaluate seven performance aspects of the mechanism.**

***Index Terms*— IoT, multicast authentication, one-way accumulator, absorbency, quasi-commutative.**

I. Introduction

IN RECENT years, Internet of Things (IoT) have developed much and gained a lot of attentions both in academia and industry [1]–[3]. By connecting sensors, tiny smart devices and everyday items with the Internet, IoT provides a new form of communications for people and devices, which makes the virtual information world integrated seamlessly with the real world [4].

According to the architecture of IoT [5], there are trillions of things being deployed in the data sensing layer, which are connected together in network access layer through different wireless communication technologies and network infrastructures (such as sensor networks, RFID systems, 3G/4G technology, and so on). For example, many sensor nodes collaborate to collect and forward data from their environment to a data management layer for processing, on the basis of which, an intelligent service layer provides various services for IoT users. In many cases, nodes are grouped together for a common goal. Accordingly, multicast is an efficient and main communication mode in both data sensing layer and network access layer of IoT.

Manuscript received January 31, 2013; revised April 29, 2013 and May 27, 2013; accepted May 28, 2013. Date of publication June 4, 2013; date of current version August 28, 2013. This work was supported in part by the Key Project of the Chinese Ministry of Education under Grant 311007 and the Chinese National High Technology Research and Development Program 863 under Grant 2012AA121604. The associate editor coordinating the review of this paper and approving it for publication was Dr. Mahmoud Daneshmand.

X. Yao, X. Han and X. Zhou are with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China (e-mail: yaoxuanxia@163.com; guangguang1225@sohu.com; xwzhouli@sina.com).

X. Du is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: dxj@ieee.org).

Digital Object Identifier 10.1109/JSEN.2013.2266116

Nowadays, more and more IoT applications have been implemented, such as environment monitoring [6], e-health [7], e-home [8] and so on. As many of them are related to user's daily life or work, the privacy and security aspect is very important. Unfortunately, the nature of the complex and heterogeneous environment in IoT makes the security issues very challenging. Furthermore, most nodes are resources-constraint, which makes the property of lightweight necessary for IoT security mechanisms.

Multicast in IoT can be easily attacked if no security mechanism is used. Hence, a lightweight authentication mechanism is needed to protect multicast messages. Multicast authentication is not only the crucial measure to ensure the authenticity of the received multicast data, but also the premise of many security services.

Many broadcast or multicast authentication mechanisms have been proposed for wireless sensor networks (WSN). These mechanisms are basis of IoT security. However, most of them can not satisfy all the seven desired properties proposed by Luk *et al*. [9]. Inspired by the properties of Nyberg's fast one-way accumulator [10], we present a multicast authentication mechanism for small scale IoT applications, which can satisfy the requirements of multicast authentications.

Our contributions are summarized below. First, we discover that the Nyberg's fast one-way accumulator has a property of absorbency in addition to the one-way and quasi-communicative properties, which makes it suitable for applications where accumulated items are dynamic. Second, we make two changes to the Nyberg's fast one-way accumulator so as to make it fit for multicast authentication and simplify the computation process. Third, we design a lightweight multicast authentication mechanism for small scale IoT applications based on the revised Nyberg's fast one-way accumulator. Fourth, we present security analysis based on probability theory and evaluate the seven performance aspects of the proposed multicast authentication scheme.

The remainder of this paper is organized as follows: In Section II, we review the related works on broadcast and multicast authentication for IoT and WSN. In Section III, we give an introduction to the one-way accumulator and Nyberg's fast one-way accumulator. In Section IV, we present our revised Nyberg's fast one-way accumulator for multicast authentication. Section V gives a description for generating signature and verification of the lightweight multicast authentication mechanism. Section VI analyzes and evaluates the security and the seven performance aspects of the pro-

posed authentication mechanism. Section VII concludes this paper.

II. Related Works

Typically, an asymmetric mechanism is required to implement multicast authentication, because the multicast environment is asymmetric and the receivers of multicast messages usually don't trust each other [9]–[12]. Current researches on multicast authentication in resource-constraint applications can be classified into three categories: public key based multicast authentication schemes; symmetric key based multicast authentication schemes; and one-time signature based authentication schemes [11].

A. Public Key Based Multicast Authentication Schemes

Public key based cryptographic mechanisms have the nature of asymmetry, which is very suitable to authenticate broadcast and multicast. However, public key based multicast authentication schemes have high computation, communication and storage overheads, which make them impractical for many resource-constraint applications, such as some IoT and WSN applications. Meanwhile, the advances in smart phones, wireless sensor nodes and other intelligent terminals as well as the developments in efficient implementation of public key cryptographic algorithms promote researches on public key based multicast authentication schemes [13]–[16].

Certificate based authentication schemes are traditional public key based multicast authentication solutions with heavy overheads. The basic or enhanced Merkle hash tree based authentication schemes [17] can balance the storage and communication overheads and improve the public key management, but the overheads are still too high for large scale applications. ID-based authentication schemes [13]–[18] are usually based on bilinear pairings, which need very high computation and energy costs and are not suitable for resource-constraint nodes in IoT. There are also some researches on pairing-free ID-based authentication scheme. The IMBAS scheme proposed by Cao *et al*. [13] is an example, which is based on a variant of BNN-IBS [18]. IMBAS has better scalability and lower energy consumption comparing to pairing ID-based authentication schemes.

In general, public key based multicast authentication schemes have some advantages in security strength and scalability, but have obvious disadvantages in overheads.

B. Symmetric Key Based Multicast Authentication Schemes

An early symmetric key based multicast authentication scheme is μ TESLA [19], based on which, there are several μ TESLA-like schemes [20]–[23]. These schemes can authenticate the source and the integrity of the multicast data by utilizing one-way hash chains to delay disclosing the authentication keys with much lower computation overheads and energy costs comparing to the public key based authentication schemes. However, they suffer from serious DoS attacks [24], [25] due to the delayed authentication [15].

In addition, the key-ring based multicast authentication scheme [26] and deployment knowledge based multicast authentication scheme [27] are all symmetric key based multicast authentication schemes. The former scheme has poor scalability and is not suitable for practical applications. The latter is also not practical due to the requirement for deployment knowledge, especially in dynamic environments, such as Internet of Things.

The symmetric key based multicast authentication schemes always construct asymmetric environment by using symmetric cryptography, time and one-way hash function or MAC. Most of them have fast computing speed, low computation overheads, communication or storage costs, but they may lack of some multicast authentication properties. Also, some special requirements for time synchronization and delayed authentications in μ TESLA-like schemes can make them vulnerable to a variety of attacks [15], [24], [25]. In addition, scalability is an issue for symmetric-key based multicast authentication schemes.

C. One-Time Signature Based Authentication Schemes

One-time signature based authentication schemes can also generate and verify signatures fast [28], [29]. Furthermore, they do not need time synchronization and can achieve authentication immediately. However, they also have some flaws, for example, the length of the signature or the verification key is too long [30]–[32], which make them only fit for applications with infrequent messages at unpredictable time [9].

In this paper, we achieve symmetric key based multicast authentication in a new way. By analyzing and revising Nyberg's fast one-way accumulator, we propose a lightweight multicast authentication mechanism, which can authenticate random multicast messages immediately at low costs for small scale applications and meet the multicast authentication requirements in resource-constraint applications.

III. Nyberg's Fast One-way Accumulator

A. One-Way Accumulator

The concept of one-way accumulator is proposed by Benaloh and Mare [33], which was designed mainly for member testing. Its general application is an alternative to digital signatures for credential authentication to verify whether one value is in a specified set or not.

For the sake of clarity, the primary notions used in Nyberg's or its revised fast one-way accumulator are listed in Table I.

A one-way accumulator is essentially a one-way hash function with the quasi-commutative property.

A function $f: X \times Y \rightarrow X$ is said to be quasi-commutative if for all $x \in X$, and for all $y_1, y_2 \in Y$,

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

If one starts with an initial value x∈X, and for all y1,y2,…yn ∈ Y, the accumulated hash value is

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

In the accumulated hashing, the accumulated list items y_i are cumulatively hashed together, and $H()$ can ensure that the accumulated hash value does not depend on the order in which the items appear on the list.

TABLE I
NOTATIONS

Notation	Meaning
H()	One-way hash function with quasi-commutative property
$H^{Nyb}()$	Nyberg's accumulating function
$H^{RNyb}()$	The revised Nyberg's accumulating function
HMAC()	HMAC function
h	One-way hash function
Z	Accumulated value
K_Z	Accumulated key
$\odot$	Logic multiplication

In the case that a one-way accumulator is used for member testing: all members keep the accumulated hash value Z, for member i, who lets z_i be the partial accumulated value of the other (n-1) member and y_i be its accumulating item. When it is required to prove that it is one of the members, it can present 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 $H(z_i,y_i) = Z$ is true, y_i is authenticated and i is one of the members.

Nowadays, one-way accumulators are usually constructed by 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 one, which is different from other one-way accumulators. It is based on the general hash function and simple bit-wise operations, therefore, fast accumulating hash 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 $m \leq N$. The one-way hash function h can map bit strings of arbitrary length to bit strings of fixed length l. For each accumulated item x_i, its hashing value is $y_i = h(x_i)$, here $i = 1,\ldots,m$. 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: $y_i = (y_{i,1},\ldots,y_{i,r})$, $y_{i,j}$ 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 comprised of d 0s, 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: $b_i = \alpha_r(y_i) = (b_{i,1},\ldots,b_{i,r})$, 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 values of r independent binary random variables, for which the probability of $b_{i,j}$ =0 can be estimated by $p(b_{i,j} = 0) = 2^{-d}$.

The accumulated hash value on Y can be computed by Equation (3).

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

For any y_i and y_j, there should be Equation (4) and (5).

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

$$H^{\text{Nyb}}\big(H^{\text{Nyb}}(K_Z, y_j), y_i\big) = K_Z \odot \alpha_r\big(h(x_j)\big) \odot \alpha_r\big(h(x_i)\big) \tag{5}$$

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

$$H^{\text{Nyb}}\big(H^{\text{Nyb}}(K_Z, y_i), y_j\big) = H^{\text{Nyb}}\big(H^{\text{Nyb}}\big(K_Z, y_j\big), y_i\big) \tag{6}$$

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

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

$$\begin{aligned} Z = H^{\text{Nyb}}(K_Z, Y) &= K_Z \odot \prod_{i=1}^{m} \alpha_r(y_i) \\ &= K_Z \prod_{i=1}^{m} \alpha_r(h(x_i)) \end{aligned} \tag{7}$$

Here, z_j is 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 $j = 1,\ldots,r$, 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 operation has the property of absorbency, which can be denoted by "$A \odot A = A$". And then Equation (8) should be true.

$$H^{\text{Nyb}}\big(H^{\text{Nyb}}(K_Z, y_i), y_i\big) = H^{\text{Nyb}}(K_Z, y_i) \tag{8}$$

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.

C. Security Analysis for 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 [10] that proves 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 $i = 1,\ldots, m (m \leq N = 2^{-d})$ and $j = 1,\ldots,r$. Let $a = (a_1,\ldots, a_r)$ be the coordinate-wise product of the r-tuples $b_i = (b_{i,1},\ldots,b_{i,r})$. Then the probability that, for all $j = 1,\ldots, r$, we have $c_j = 0$ only i.f $a_j = 1$, is equal to $2^{-d}(1\text{-}2^{-d})^m$.

It is assumed that h is an ideal random one-way hash function. If we consider c as the result that an forged accumulated item has been hashed and mapped by rule α_r, then for each $j = 1,\ldots,r$, the probability that $c_j = 0$ and $a_j = 1$ is $P_f =$

$(1\text{-}2^{-d}(1\text{-}2^{-d})^m)^r$, which is also the probability of forging an accumulated item successfully.

Since $N = 2^d$, there is $d = \log N$, accordingly, Equation (9) should be true:

$$P_f = (1 - 2^{-d}(1 - 2^{-d})^m)^r = \left(1 - \frac{1}{N}\left(1 - \frac{1}{N}\right)^m\right)^r \quad (9)$$

Here, due to $m \leq N$, there is Inequality (10).

$$P_f \leq \left(1 - \frac{1}{N}\left(1 - \frac{1}{N}\right)^N\right)^r \quad (10)$$

When $N \to \infty$, $(1-\frac{1}{N})^N \to e^{-1}$, Where e is Neper's number. And then Inequality (10) can be expressed by Inequality (11).

$$P_f \leq \left(1 - \frac{1}{N}\left(1 - \frac{1}{N}\right)^N\right)^r \approx \left(1 - \frac{1}{N \times e}\right)^r \approx e^{\frac{r}{N \times e}} \quad (11)$$

Let $t = \frac{r}{N \times e}$, there is Equation (12):

$$r = N \times e \times t \quad (12)$$

Here, t is called security level. According to Inequality (11), when t is big enough, P_f is small enough and the security of Nyberg's fast one-way accumulator is strong enough.

The length of the hash code l is given by Equation (13):

$$l = r \times d = N \times e \times t \times d = 2^d \times e \times t \times d \quad (13)$$

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 the security level t be 10, then the length of the hash code is 278 Kbits, which is much longer than the hash code (128-512 bits) of existing hash functions. A simple way to get the required long hash code is hashing the accumulated item to a short hash code first, and then using the short hash code as a seed for a binary random sequence generator. From this seed, many pseudorandom bits can be generated for the long hash code.

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

In general, a one-way accumulator can be used to mutual-authenticate members of one group made up of fixed items. 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 for the applications in which the accumulated items are dynamic. These characteristics inspire us to apply it to multicast authentication. In [34], we use MAC as the accumulated items of the Nyberg's fast one-way accumulator for multicast authentication. In this paper, we revise Nyberg's fast one-way accumulator to further simplify the computing process in multicast authentication.

In order to take the advantage of Nyberg's fast one-way accumulator's virtues in computing, we still exploit MAC to construct the multicast authentication mechanism. The difference from literature [34] is that the MAC computing process is embedded in Nyberg's fast one-way accumulator, which can achieve multicast authentication on multicast data directly. For this purpose, we make two changes on Nyberg's fast one-way accumulator.

1) The accumulated item is changed from a single data to a two-tuple, which means each accumulated item is made up of two elements, one of which is the shared key between the multicast source and the destination, the other is the multicast data. The difference between two accumulation items is the shared key, which is only known to the sender and the corresponding receiver.
2) Since a HMAC function also has the one-way property as a hash function, a HAMC function is used to substitute the hash function h in Nyberg's fast one-way accumulator so as to embed the MAC's computing process into Nyberg's fast one-way accumulator.

Consequently, the revised Nyberg's fast one-way accumulator can be given as Equation (14).

$$\begin{aligned} Z(Y) &= H^{RNyb}(K_Z, Y) = K_Z \odot \alpha_r\big(HMAC(X)\big) \\ &= K_Z \odot \prod_{i=1}^{m} \alpha_r(y_i) \\ &= K_Z \odot \prod_{i=1}^{m} \alpha_r\Big(HMAC(K_{S,j}, Data)\Big) \end{aligned} \quad (14)$$

The security of the revised Nyberg's fast-one way accumulator is similar to the original one, which eventually depends on the randomness properties of the HMAC function. Its security analysis is the same as the original Nyberg's fast-one way accumulator. 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 revised Nyberg's fast one-way accumulator, we can conveniently achieve multicast authentication. The details are given in Section V.

V. The Lightweight Multicast Authentication Mechanism for Small Scale IoT Applications

A. Application Model

In resource-constraint systems, such as WSN, IoT, and ubiquitous computing environment, multicast is a common means of communications. In order to present the multicast authentication process, some assumptions of the applications are made as following.

1) All nodes in the network can use the revised Nyberg's fast one-way accumulator to perform the accumulation operation.
2) A node can establish a unique shared key with each of its neighbor using an existing key management scheme.
3) A node can establish a unique shared key with each node in its group by using an existing key management scheme.
4) There is a group key in a group. The group key is known by all members in the group.

B. The Multicast Authentication Process

The revised Nyberg's fast one-way accumulator uses the following two things as the accumulated items: 1) the shared

keys between the sender and the destinations; and 2) the multicast data. And it uses *HMAC*() to do hashing. Thereof, both the private information (i.e., the shared keys) and the multicast data are embedded into the accumulated value, which serves as the signature of the multicast source. If the signature passes the receiver's verification, it indicates that: 1) the two-tuple is made up of the shared key between the receiver and the multicast source; and 2) the multicast data is an accumulated item of the accumulated value. Consequently, both the multicast source and the data are all authenticated. This means that the multicast message comes from the claimed source and the data is unchanged.

To present the procedure of the proposed mechanism, the following notations are used:

$K_{i,j}$: the shared key between node i and node j
S: multicast source
R_i: multicast receiver i
ID_j: the ID of node j
Z: accumulated value computed by the multicast source
Z': signature received by a node
Data: multicast data
M: multicast message, $M =$"$ID_s \| Data \| Z$"
($K_{S,i}$, *Data*):accumulated item

We use the following example to describe the multicast authentication process: Node S sends multicast data to all its neighbors or all other nodes in a certain group. It is assumed that the number of its neighbors or group members is $(m+1)$. To generate a signature for the multicast data, three steps should be done by node S.

Step1. Node S constructs the accumulated item set. For this purpose, it should use the shared key with each scheduled receiver and the multicast data to form the two-tuple ($K_{S,i}$, *Data*) set.

Step2. Node S computes the accumulated value on each two-tuple ($K_{S,i}$, *Data*) in the accumulated item set according to Equation (15). Generally, the accumulated key K_z can be randomly selected. But for the security of multicast authentication (the analysis is given in Section VI), there must be enough bits of 1 in the accumulated value or signature, we make $K_z = \{1\}^r$.

$$Z = K_Z \odot \prod_{i=1}^{m} \alpha_r \Big(HMAC(K_{S,i}, Data) \Big) \tag{15}$$

Step 3. Multicast message M is constructed and sent. Here $M =$"$ID_S \| Data \| Z$".

When node i receives such a multicast message M, five steps are needed for authentication.

Step 1. The identifier of the claimed multicast source S, *Data* and the received signature Z' (which may be different from the sent-out value Z) should be extracted from M, respectively.

Step 2. Node i checks whether the number of "1" bits in signature Z' meets the requirements (the reason is given in Section VI) or not. If not, the multicast is considered forged and should be discarded, else go to step 3.

Step 3. Node i checks if it is a neighbor of or in the multicast source's group according to ID_S. If yes, go to step 4. Otherwise, node i is not a scheduled receiver, and it will either forward this message or discard it according to the policy.

Step 4. Node i uses the shared key $K_{S,i}$ 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 5. Node i performs logical multiplication operation on b_i and the received signature Z'. If $Z' = Z' \odot b_i$, it indicates that this multicast message is from node S and its multicast data is not tampered. Otherwise, it indicates that the multicast data has been tampered or the multicast message is not from node S.

In addition, if the multicast destinations are the members in a group and the confidential property is required, we can use the group key to encrypt the multicast data and make the encrypted multicast data as the second part of the accumulated item. In this way, we can multicast confidential data and authenticate it. The multicast process and authentication process are the same as the situation where confidential property is not required.

VI. Analysis of the Multicast Authentication Mechanism

A. Security Analysis

For the convenience of analysis, the security level t in subsection C of Section III is adopted here and the event with a probability less than e^{-t} is considered as impossible at security level t. A successful attack on the multicast authentication mechanism 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}, it is considered secure at t level.

It is assumed that the randomness of the HMAC function can be guaranteed. Since the multicast algorithm is constructed on the revised Nyberg's fast-one way accumulator, the security of it should be analyzed on the basis of meeting the security requirements of the revised Nyberg's fast one-way accumulator, which means that the length of the accumulation 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 multicast authentication mechanism is a reverse process. A receiver authenticates the multicast message by checking whether the "$Z' \odot b_i = Z'$" is true or not, where Z' is the received signature by node i, and b_i is calculated by Equation (16).

$$b_i = \alpha_r \Big(HMAC\left(K_{S,i}, Data\right) \Big) \tag{16}$$

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. One success attack means that a forged signature can make a receiver believe 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 $Z' = \{0\}^r$, the Equation $Z' \odot b_i = Z'$ 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 distribution of the value of each bit in an accumulated value Z satisfies the binomial distribution. Assume there are $m(m \leq N)$ accumulation items. Let Z_i be the i^{th} bit in Z, we have two Equations:

$$P(Z_i = 1) = (1 - 2^{-d})^m \tag{17}$$

$$P(Z_i = 0) = (1 - (1 - 2^{-d})^m \tag{18}$$

Accordingly, the probability that there are q bits "1" in Z can be estimated by Equation (19):

$$\mathrm{P}(q) = \binom{r}{q} \left(\left(1 - 2^{-d}\right)^m \right)^q \cdot \left(1 - (1 - 2^{-d})^m\right)^{(r-q)} \tag{19}$$

It can be seen from Equation (19) that q hinges on r, and r 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 Equation (20), where k is between 0 and r.

$$\begin{aligned} \mathrm{P}(q > k) &= 1 - \mathrm{P}(q \leq k) \\ &= 1 - \sum_{i=0}^{k} (ri) \left(\left(1 - 2^{-d}\right)^m\right)^i \left(1 - \left(1 - 2^{-d}\right)^m\right)^{(r-i)} \end{aligned} \tag{20}$$

According to Equation (20), we can see that the probability of $(q>k)$ 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 in advance, so the minimum of q only depends on r. Of course, the minimum q can be known in advance for a given application, and the number of "1" in the signature can be set as not less than the minimum of q, which makes attacks more difficult, and also helps a receiver distinguish a forged multicast quicker. For instance, let the number of "1" in the received signature (it may be a forged one) be $q\prime$, a multicast with $q\prime<\min(q)$ is considered as fraudulence and not necessary to be further verified. Only a multicast with $q\prime \geq \min(q)$ needs to be further authenticated.

Any receiver R_i computes $HMAC()$ on the received multicast data using the shared key between itself and the claimed sender and perform permutation on it to get b_i. The probability of $b_{i,j} = 0$ can be estimated by $\mathrm{P}\{b_{i,j} = 0\}=2^{-d}$, and the probability of $b_{i,j} = 1$ can be estimated by $\mathrm{P}\{b_{i,j} = 1\}= 1\text{-}2^{-d}$. The probability of any bit in the received signature being 1 can be estimated by $\mathrm{p}(Z'_j = 1)= q\prime/r$. The probability of cheating a receiver to trust the signature can be estimated by Equation (21):

$$\mathrm{P_F} = \left(1 - (2^{-d} \times q'/r)\right)^r = \left(1 - \left(q'/(N \times r)\right)\right)^r \leq e^{-q'/N} \tag{21}$$

Equation (21) indicates that probability of a success attack depends on $q\prime$ and N. When $(N\times r)/q\prime$ tends to infinity, it equals $e^{-\mathrm{q}\prime}/\mathrm{N}$. If N is given, it only depends on $q\prime$, and $q\prime$ further depends on r. So we can enlarge r to decrease the probability of attack successfully or ensure the security of algorithm. Let security level $t = 10$, we can compute minimum q according to Equation (21). And then r can be calculated by Equation (20).

TABLE II
RELATIONS BETWEEN THE LENGTH OF SIGNATURE AND THE NUMBER OF SCHEDULED RECEIVER

d	N	Minimum q (bits)	Minimum r (bits)
2	4	40	208
3	8	80	320
4	16	160	576
5	32	320	1056

The values of Minimum q and Minimum r are computed when the security level is 10.

N is the maximum number of the scheduled receivers, and $\mathrm{N} = 2^{\mathrm{d}}$.

When $N = 4,8,16,32$, the values of minimum q and r are shown in Table II. For the sake of simplicity, the value of r is the integer times of one byte (8 bits).

We can see from Table II that the length of signature r increases sharply with the increment of the number of the accumulated items when t is fixed, and the cost of communication will increase with the increment of r, which indicates that the multicast mechanism based on the revised Nyberg fast one-way accumulator is not fit for large scale multicast applications but very suitable for small scale applications with limited resources that are not suitable to use public key algorithm to signature. For instance, sensor nodes in wireless sensor network multicast to their neighbors or the cluster head multicast to the nodes in its cluster and so on.

B. Performance Evaluation

The performance of the proposed multicast authentication mechanism is evaluated on seven aspects according to [9].

1) Resistance Against Node Compromise: Since nodes in most IoT applications are not equipped with tamper-resistant hardware and often deployed in unattended environments, any captured nodes will result in the disclosure of confidential information, which will further affect the security of other nodes and the communications among them. The resistance against node compromise refers to the degree of effects on other nodes' security when a node is compromised.

In the proposed scheme, the signature is generated by the multicast data and each shared key between the multicast source and every scheduled receiver, and the shared key is only known to the multicast source and the scheduled receiver. If one scheduled receiver is compromised, it will not affect other scheduled receivers to verify the signature and their communication. Only the communications between the comprised node and the nodes that have sharing confidential information with it become insecure.

2) Low Computation Overhead: Compared to the signature based on public key system (such as RSA, and ECC), the computation overhead of the proposed scheme is very low.

TABLE III
THE PERFORMANCE COMPARISON WITH SOME EXISTING MULTICAST AUTHENTICATION MECHANISMS

	Resistance Against Node Compromise	Computation Overhead	Communication Over-head(Bytes)	Robustness to Packet Loss	Immediate Authentication	Message Sent at Irregular Times	High Message Entropy
One MAC based multicast authentication	×	2	20	√	√	√	√
μ TESLA	√	3	40	√	×	√	√
RPT(Regular-Predictable TESLA)	√	2	40	√	√	×	√
One-Time Signature	√	268	460	√	√	√	×
HORS	√	2	>1000K	×	√	√	√
Public-key based signature	√	>1000	40	√	√	√	√
The proposed scheme	√	N+1	$\leq$ 132	√	√	√	√

"√" indicates the property is supported by the algorithm.

"×"indicates the property is not supported by the algorithm.

N is the maximum number of the scheduled receivers, it is assumed to be not more than 32.

The ECC based signature algorithm is taken as an example of public-key based signature, and it is assumed the length of the key is 160 bits.

One-time signature is assumed to sign on a 80-bit data using the chains of length 16. The computation overhead of it is an average value.

It is assumed that HORS use the suggested parameter values in [31].

Here, the main operations of the signature generation and verification can be classified into three types: logical multiplication, *HMAC*(), and bits mapping. According to [14], even using the advanced ECC signature schemes, the overhead of signature generation is 4 orders of magnitude slower than the MAC generation, and the overhead of signature verification is 3 orders of magnitude slower than the MAC verification. In the proposed scheme, the main computation overhead for signature generation and verification is $(N+1)$ HMAC operations. For small-scale multicast applications (e.g., only dozens of receivers), the computation overhead is much lower than that of digital signature. In addition, logical multiplication and bits mapping are all simple bit operations in nature. Hence, the proposed multicast authentication mechanism improves execution efficiency and is suitable for resource-constraint situations.

3) Low Communication Overhead: The communication overhead depends on the length of the message. Most multicast authentication messages are composed of the ID of the multicast source, data, the signature and the verification key. The first two parts are prerequisite and the length of them can hardly vary. The verification key is an optional item and can be transmitted separately. Based on this, the communication overhead can be evaluated by the length of the signature and the required verification key. In the proposed multicast authentication scheme, the length of the signature depends on the number of scheduled receivers (see Table II) and the verification key does not need to be transmitted. In a small-scale multicast application, such as only 16 or fewer scheduled receivers, the length of the signature is no more than 72 bytes, which is longer than that of μTESLA but much less than that of key-ring based scheme, in which the length of the signature is the product of the length of the key ring and the length of one MAC. In large scale applications, the signature's length will increase sharply with the number of the receivers and cause high communication overhead.

4) Robustness to Packet Loss: Robustness to packet loss belongs to the issue of network reliability. Here it is used to indicate that an attacker can't forge a signature if some packets are dropped. In the proposed algorithm, if a multicast message is not fragmented, each multicast packet is independent of each other. If the multicast data is fragmented, several multicast packets are used for one multicast data. Since different parts of the multicast data are signature and sent independently, and there is no security relationship among them, they can also be considered as independent multicast packets. For a receiver, one packet loss can only cause not receiving that message or not able to recover the message under the circumstance of fragmentation, which do not affect other receivers and multicast messages. For an attacker, one or more packet loss does not help him forge a signature, because he can't find out any secret information from packet loss. In fact, the robustness can be guaranteed in most multicast authentications except for HORS[31].

5) Immediate Authentication: Most multicast authentication mechanisms based on μTESLA use the asymmetry in time among the broadcast source and receivers through a delayed disclosure of keys, which causes authentication delay.

The proposed scheme constructs the asymmetric mechanism based on the one-way accumulator and does not depend on the asymmetry in time or key chaining. A multicast message can include all the information required by the process of authentication whether the multicast data is fragmented or not. Hence, a scheduled receiver can authenticate the multicast data immediately as soon as it receives the entire multicast message.

6) Messages Sent at Irregular Times: The property of messages sent at irregular times is an essential attribute for almost all multicast authentication schemes. Schemes that require messages sent at regular times are designed to meet the requirement of certain specified applications. The proposed

multicast authentication mechanism doesn't depend on time factor, and each multicast message is independent. Hence, a source can multicast to its neighbors (or members in the same group) at any time.

7) High Message Entropy: All the existing multicast authentication schemes (except the one-time signature) have the property of high message entropy. The proposed multicast authentication mechanism does not set any constraints on the multicast data and the entropy of the multicast data has nothing to do with the signature, which make it authenticate messages with high entropy.

Let the computation overhead for one HMAC or hash to be a unit, the length of the signature and the key that should be transmitted for authentication to be the metric of communication overhead. The computation overheads include the overheads more than a unit in signature generation and signature verification. It is assumed that the length of the output of a HMAC or hash is 160 bits. The comparisons in the seven aspects between the proposed multicast authentication mechanism and some existing multicast authentication mechanisms are shown in Table III.

VII. Conclusion

Multicast is a main communication mode in the Internet of things (IoT). To achieve secure and efficient multicast in IoT, lightweight multicast authentication is required. Inspired by the absorbency property of the original Nyberg's fast one-way accumulator, we present a revised Nyberg's fast one-way accumulator and construct a lightweight multicast authentication algorithm for small scale resource-constraint IoT applications. We analyzed the security property in detail. In addition, we evaluated the seven cardinal properties that are required by multicast authentications for resource-constraint applications. The analysis results showed that the proposed multicast authentication algorithm meets the requirements of resource-constraint applications.

References

[1] R. Roman, P. Najera, and J. Lpoez, "Securing the internet of things," *Comput.*, vol. 44, no. 9, pp. 51–58, Sep. 2011.

[2] K. Främling and J. Nyman. (2008). *The Compromise Between Security and Usability in the Internet of Things* [Online] Available: http://www.cs.hut.fi/~framling/Publications/FramlingNymanAPMS2008_Final-17-05-2008.pdf

[3] R. H. Weber, "Internet of things—New security and privacy challenges," *Comput. Law Security Rev.*, vol. 26, no. 1, pp. 23–30, Jan. 2010.

[4] D. Bandyopadhyay and J. Sen, "Internet of things: Applications and challenges in technology and standardization," *Wireless Pers. Commun.*, vol. 58, no. 1, pp. 49–69, May 2011.

[5] D. Chen, G. Chang, L. Jin, X. Ren, J. Li, and F. Li, "A novel secure architecture for the internet of things," in *Proc. 5th Int. Conf. Genetic Evol. Comput.*, 2011, pp. 311–314.

[6] N. Dlodlo, "Adopting the internet of things technologies in environmental management in South Africa," in *Proc. Int. Conf. Environ. Sci. Eng.*, vol. 3. Apr. 2012, pp. 45–55.

[7] J. Li, X. Wu, and H. Chen, "Research on mobile digital health system based on internet of things," in *Electrical Power Systems and Computers* (Lecture Notes in Electrical Engineering), vol. 99. New York, NY, USA: Springer-Verlag, 2011, pp. 495–502.

[8] X. Shang, R. Zhang, and Y. Chen, "Internet of things (IoT) services: Architecture and its application in E-commerce," *J. Electron. Commerce Org.*, vol. 10, no. 3, pp. 44–55, Jul.–Sep. 2012.

[9] M. Luk, A. Perrig, and B. Whillock, "Seven cardinal properties of sensor network broadcast authentication," in *Proc. 4th ACM Workshop Security Ad Hoc Sensor Netw.*, 2006, pp. 147–156.

[10] K. Nyberg, "Fast accumulated hashing," in *Proc. 3rd Fast Softw. Encrypt. Workshop*, 1996. pp. 83–87.

[11] X. Fan and G. Gong, "Accelerating signature-based broadcast authentication for wireless sensor networks," *Ad Hoc Netw.*, vol. 10, no. 4, pp. 723–736, Jun. 2012.

[12] D. Boneh, G. Durfee, and M. Franklin, "Lower bounds for multicast message authentication," in *Proc. Adv. Cryptol.*, 2001, pp. 434–450.

[13] X. Cao, W. Kou, L. Dang, and B. Zhao, "IMBAS: Identity-based multi-user broadcast authentication in wireless sensor networks," *Comput. Commun.*, vol. 31, no. 4, pp. 659–67, 2008.

[14] D. Malan, M. Welsh, and M. Smith, "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography," in *Proc. 1st Annu. IEEE Commun. Soc. Conf. Sensor Ad Hoc Commun. Netw.*, Oct. 2004, pp. 71–80.

[15] K. Ren, S. Yu, W. Lou, and Y. Zhang, "Multi-user broadcast authentication in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 8, pp. 4554–4564, Oct. 2009.

[16] S. Yamakawa, Y. Cui, K. Kobara, and H. Imai, "Lightweight broadcast authentication protocols reconsidered," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2009, pp. 3076–3081.

[17] D. Liu, P. Ning, S. Zhu, and S. Jajodia, "Practical broadcast authentication in sensor networks," in *Proc. 2nd Annu. Int. Conf. Mobile Ubiquitous Syst., Netw. Services*, Jul. 2005, pp. 118–129.

[18] M. Bellare, C. Namprempre, and G. Neven, "Security proofs for identity-based identification and signature schemes," in *Proc. Adv. Cryptol.*, vol. 3027. 2004, pp. 268–286.

[19] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security protocols for sensor networks," in *Proc. 7th Annu. ACM Int. Conf. Mobile Comput. Netw.*, Jul. 2001, pp. 189–199.

[20] K. Ren, K. Zeng, W. Lou, and P. J. Moran, "On broadcast authentication in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 11, pp. 4136–4144, Nov. 2007.

[21] D. Liu and P. Ning, "Multi-level μTESLA: Broadcast authentication for distributed sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 3, no. 4, pp. 800–836, 2004.

[22] D. Liu, P. Ning, S. Zhu, and S. Jajodia, "Practical broadcast authentication in sensor networks," in *Proc. MobiQuitous*, 2005, pp. 118–129.

[23] J. Shaheen, D. Ostry, V. Sivaraman, and S. Jha, "Confidential and secure broadcast in wireless sensor networks," in *Proc. 18th Int. Symp. Pers., Indoor, Mobile Radio Commun.*, 2007, pp. 1–5.

[24] Q. Dong, D. Liu, and P. Ning, "Pre-authentication filters: Providing DoS resistance for signature-based broadcast authentication in wireless sensor networks," in *Proc. 1st ACM Conf. Wireless Netw. Security*, 2008, pp. 2–12.

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

[26] T. Wu, Y. Cui, B. Kusy, A. Ledeczi, J. Sallai, N. Skirvin, J. Werner, and Y. Xue. (2007) *A Fast and Efficient Source Authentication Solution for Broadcasting in Wireless Sensor Networks* [Online]. Available: http://www.stanford.edu/~kusy/pubs/ntms07-wu-fast.pdf

[27] H. Lee, D. Nyang, and J. Song, "Message and its origin authentication protocol for data aggregation in sensor networks," in *Proc. Emerg. Direct. Embedded Ubiquitous Comput.*, 2006, pp. 281–290.

[28] A. Perrig, "The BiBa one-time signature and broadcast authentication protocol," in *Proc. 8th ACM Conf. Comput. Commun. Security*, 2001, pp. 28–37.

[29] S. Chang, S. Shieh, W. Lin, and C.-M. Hsieh, "An efficient broadcast authentication scheme in wireless sensor networks," in *Proc. ACM Symp. Inf., Comput. Commun. Security*, 2006, pp. 311–320.

[30] R. Merkle, "A digital signature based on a conventional encryption function," in *Proc. Adv. Cryptol.*, 1988, pp. 369–378.

[31] L. Reyzin and N. Reyzin, "Better than BiBa: Short one-time signatures with fast signing and verifying," in *Proc. 7th Australasian Conf.*, Jul. 2002, pp. 144–153.

[32] P. Rohatgi, "A compact and fast hybrid signature scheme for multicast packet," in *Proc. 6th ACM Conf. Comput. Commun. Security*. Nov. 1999, pp. 93–100.

[33] J. Benaloh and M. de Mare, "One-way accumulators: A decentralized alternative to digital signatures," in *Proc. Adv. Cryptol.*, 1993, pp. 274–285.

[34] X. Yao, X. Zheng, and X. Zhou, "Broadcast authentication algorithm for wireless sensor networks," *J. China Inst. Commun.*, vol. 31, no. 11, pp. 49–55, Nov. 2010.

Xuanxia Yao received the B.S. degree in computer application from Jiangsu University, Zhenjiang, China, and the M.S. and Ph.D. degrees in computer application from the University of Science and Technology Beijing (USTB), Beijing, China, in 2002 and 2009, respectively. She is a member of China Computer Federation.

From 1994 to 1999, she was a Research Assistant with the Computer Center, Luoyang Mining Machinery Institute of Technology. Since 2009, she has been an Associate Professor with the School of Computer and Communication Engineering, USTB. She is the author of one book and more than 20 articles. Her current research interests include networks and information security, wireless sensor networks, Internet of Things, and cloud computing.

Xiaoguang Han received the B.S. degree in computer application technologies from the Handan Institute of Technology, Handan, China, in 2008. He is currently pursuing the Ph.D. degree with the School of Computer & Communication Engineering, University of Science & Technology Beijing, Beijing, China.

He is the author of five articles. His current research interests include cloud computing, network security, and privacy protection.

Xiaojiang Du (SM'03) received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1996 and 1998, respectively, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, USA, in 2003.

He was an Assistant Professor with the Department of Computer Science, North Dakota State University, Fargo, ND, USA, in August 2004 and July 2009, where he received the Excellence in Research Award in May 2009. He is currently an Associate Professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA. His current research interests include security, cloud computing, wireless networks, and computer networks and systems. He has published over 100 journal and conference papers. He is a Life Member of ACM.

Xianwei Zhou received the B.S. degree in mathematics from Southwest Normal University, Chongqing, China, in 1986, the M.S. degree in mathematics from Zhengzhou University, Zhengzhou, China, in 1992, and Ph.D. degree in communication engineering from Southwest Jiaotong University, Shanghai, China, in 1996.

He is a Professor with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China. He is the author of five books and more than 100 articles. His current research interests include smart networks, information security, and Cyber security.