

# **AN AUTHENTICATION FOR GROUP COMMUNICATIONS IN AD-HOC NETWORKS.**

**Alabi Peter Akubo**

Department of Information Technology, Federal University of Technology, (FUTO) Owerri.

[peterakubo@gmail.com](mailto:peterakubo@gmail.com) & akubo.pa@hotmail.com

## **ABSTRACT**

*Group communication implies a many-to-many communication and it goes beyond both one-to-one communication (i.e., unicast) and one-to-many communication (i.e., multicast). Unlike most user authentication protocols that authenticate a single user each time, we propose a new type of authentication, called **group authentication** that authenticates all users in a group at once. The group authentication protocol is specially designed to support group communications. There is a group manager who is responsible to manage the group communication. During registration, each user of a group obtains an unique token from the group manager. Users present their tokens to determine whether they all belong to the same group or not. The group authentication protocol allows users to reuse their tokens without compromising the security of tokens. In addition, the group authentication can protect the identity of each user.*

**KEYWORDS:** *User authentication; Group communication; secret sharing; Ad hoc network; Strong t - consistency*

## **1. INTRODUCTION**

User authentication is one of the most important security services in computer and communication application. Knowledge based authentication (e.g., password) [16, 9] and key based authentication (e.g., public/private key) [7,12] are the two most popular approaches. Knowledge based authentication has some security flaws. Most users like to use simple and short passwords. However, Internet hackers can easily crack simple passwords. Public-key based authentication needs a certificate authority (CA) to provide the authenticity of public keys. In addition, public-key computations involve large integers. Computational time is one of the main concerns for public-key based authentication.

All user authentication protocols [10,6] are one-to-one type of authentication where the prover interacts with the verifier to prove the identity of the prover. For example, the RSA digital signature [13] is used to authenticate the signer of the signature. In this approach, the verifier sends a random challenge to the prover. Then, the prover digitally signs the random challenge and returns the digital signature of the challenge to the verifier. After successfully verifying the digital signature, the verifier is convinced that the prover is the one with the identity of the public key used to verify the digital

signature. In wireless communications, when a mobile subscriber wants to establish a connection with the base station, the subscriber and the base station interact to DOI: 10.5121/ijnsa.2013.5302  
09 establish mutual authentication. Mutual authentication can prevent an illegitimate subscriber from using the service and prevent the fake base station from harming the subscriber.

Network applications are no longer just one-to-one communication; but involve multiple users (□ 2). Group communication [14,2] implies a many-to-many communication and it goes beyond both one-to-one communication (i.e., unicast) and one-to-many communication (i.e., multicast). In this paper, we propose a new type of authentication, called **group authentication**, which authenticates all users in a group at once. The group authentication protocol is specially designed to support group communications. The group authentication is defined to involve multiple users and users want to convince each other that they all belong to the same group without revealing their identities. In the group authentication, each user acts as both the prover and the verifier. Group authentication is extremely important in an ad hoc network because this network is temporarily established by multiple users and these users want to use this network to exchange secret information.

Devising protocols to provide group authentication in ad hoc networks is extremely challenging due to highly dynamic and unpredictable topological changes. As a result, there are two popular models to provide group authentication services in an ad hoc network. The first model involves a centralized authentication server (AS) [11,3] and the second model has no AS [5,4]. In the first model, AS manages the access rights of the network. For example, Bhakti et al. [3] proposed to adopt Extensible Authentication Protocol (EAP) in the IEEE 802.1x standard for wireless ad hoc network. This approach requires to set up the AS and have mobile users to access to the AS service. In fact, in some situations, the second model is the only way to provide group authentication. For example, in an ad-hoc network communication, there has no AS service available to mobile users. In the second model, each user needs to take in charge of authenticating other users. In a straightforward approach, if there are  $n$  users in the group, each user can use the one-to-one authentication protocol for  $n - 1$  times to authenticate other users. Computational time is one of the major concerns in this approach.

In this paper, we introduce a special type of group authentication which provides an efficient way to authenticate multiple users belonging to the same group without revealing identity of each user. Our proposed protocol is no longer a one-to-one type of authentication. It is a many-to-many type of authentication. Unlike most user authentication protocols that authenticate a single user each time, our proposed protocol authenticates all users of a group at once. In our proposal, each user needs to register with a group manager (GM) to become a group user. Like the trusted dealer in Shamir's  $(t, n)$  secret sharing scheme [15], the GM needs to select a secret polynomial and compute token for each user. Based on these tokens, our protocol can establish group authentication for all users at once. The group authentication protocol allows users to reuse their tokens without compromising the security of tokens. Our proposed protocol supports existing wireless communication network including wireless ad hoc network.

The rest of this paper is organized as follows. In next section, we include some preliminaries. In Section 3, we introduce the model of our proposed group authentication. In Section 4, we present basic one-time group authentication protocol; in Section 5, we present group authentication protocol without revealing tokens. We conclude in Section 6.

## 1. PRELIMINARIES

### 2. Review of Shamir's secret sharing scheme [15]

In Shamir's  $(t, n)$  secret sharing scheme based on the polynomial, there are  $n$  shareholders and a mutually trusted dealer. The scheme consists of two algorithms:

- a) **Share generation algorithm:** the dealer first picks a random polynomial of degree  $t - 1$ ,  $f(x) = a_{t-1}x^{t-1} + \dots + a_1x + a_0 \pmod{p}$ , such that the secret  $s$  satisfies  $f(0) = a_0 = s$  and all coefficients,  $a_0, a_1, \dots, a_{t-1} \in \mathbb{Z}_p$ ,  $p$  is a prime with  $p > s$ . The dealer computes shares,  $f(x_i)$ , for  $i = 1, 2, \dots, n$ , and distributes each share  $f(x_i)$  to shareholder  $U_i$  secretly.
- b) **Secret reconstruction algorithm:** it takes any  $t$  or more than  $t$  shares, for example,  $j$  shares (i.e.,  $t \leq j \leq n$ ),  $(x_1, f(x_1)), (x_2, f(x_2)), \dots, (x_j, f(x_j))$ , as inputs, and outputs the secret  $s$  using Lagrange interpolating formula as

$$s = \sum_{i=1}^j f(x_i) \prod_{r=1, r \neq i}^j \frac{x - x_r}{x_i - x_r} \pmod{p}.$$

We note that the above algorithms satisfy the basic requirements of the secret sharing scheme, that are, (1) with the knowledge of any  $t$  or more than  $t$  shares, shareholders can reconstruct the secret  $s$ ; and (2) with the knowledge of any  $t - 1$  or fewer than  $t - 1$  shares, shareholders cannot obtain the secret  $s$ . Shamir's secret sharing scheme is unconditionally secure since the scheme satisfies these two requirements without making any computational assumption. For more information on this scheme, please refer to the original paper [15].

#### Harn and Lin's definition on strong $t$ -consistency [8]

Benaloh [1] presented a notion of  $t$ -consistency to determine whether a set of shares is generated from a polynomial of degree  $t - 1$  at most. Recently, Harn and Lin [8] proposed a new definition of strong  $t$ -consistency which is the extension of Benaloh's definition.

**Definition 1 (Strong  $t$ -consistency [8]).** A set of  $n$  shares (i.e.,  $t < n$ ) is said to be strong  $t$ -consistent if (a) any subset of  $t$  or more than  $t$  shares can reconstruct the secret, and (b) any subset of fewer than  $t$  shares cannot reconstruct the secret.

It is obvious that if shares in Shamir's secret sharing scheme are generated by a polynomial with degree  $t - 1$

exactly, then shares satisfy the security requirements of a  $(t, n)$  secret sharing scheme and these shares are also strong  $t$ -consistent.

Checking strong  $t$ -consistency of  $n$  shares can be executed very efficiently by using Lagrange interpolating formula. In fact, to check whether  $n$  shares are strong  $t$ -consistent or not, it only needs to check whether the interpolation of  $n$  shares yields a polynomial with degree  $t - 1$  exactly. If this

condition is satisfied, we can conclude that all shares are strong  $t$ -consistent. However, if there are some illegitimate shares, the degree of the interpolating polynomial of these  $n$  shares is more than  $t$  with very high probability. In other words, these  $n$  shares are most likely to be not strong  $t$ -consistent. The property of strong  $t$ -consistency will be used in Section 5 of our protocol to check strong  $t$ -consistency of  $n$  shares without revealing tokens.

## 1. MODEL

### Entities

- a) **Group Manager (GM):** A group manager is responsible to register users to form a group. The responsibility of GM is to issue a secret token to each user during registration. Later, authentication is based on the secret tokens. Since tokens are used in authentication, identities of users are protected. In order to prevent malicious users to reveal their tokens to attackers, each token is a unique integer. The secret tokens are shares of the polynomial generated by the GM.
- b) **Group Users:** Join a group and become a group user, each user needs to register with the GM. After being successfully registered, each user receives a secret token from the GM. Each user with a unique token can prevent malicious users to give their tokens to impersonators.
- c) **Attackers:** We consider two types of attackers, the *inside attackers* and the *outside attackers*. The inside attackers are users who are legitimate users and own legitimate tokens from the GM. We consider that the insider attackers may collude to forge tokens for non-users. The outside attackers are impersonators who do not own any tokens and try to impersonate users to fail the authentication protocol. We also assume that the GM does not collude with any user. If the GM colludes with any user by revealing the secret of the GM to the user, the colluded user can do harm to the group. In addition, we assume all users act honestly in the authentication. If any user acts dishonestly by revealing an invalid value, the authentication is failed.

### Authentication outcomes

There are only two possible outcomes of a group authentication; that are, either “yes” or “no”. If the outcome is “yes”, it means that all users belong to the same group; otherwise, there are impersonators.

## 2. BASIC ONE-TIME GROUP AUTHENTICATION PROTOCOL

In the following discussion, we assume that there are  $n$  users,  $M_1, M_2, \dots, M_n$ , registered at the GM to form a group.

### System set up

During registration, GM constructs a random  $(t - 1)$ -th (i.e.,  $t < n$ ) degree polynomial

$f(x)$  with

$f(0) = \sigma$ , and computes secret tokens of users as

$y_i = f(x_i)$ , for  $i = 1, 2, \dots, n$ , where  $x_i$

is the public information associated with user  $M_i$ . GM sends each token GM makes  $H(s)$  publicly known, where  $H$  is a one-way function.

$y_i$  to user  $M_i$  secretly.

*Remark 1.* The threshold  $t$  is an important security parameter that affects the security of group authentication protocols. Using a  $(t, n)$  secret sharing scheme to issue tokens in the registration can prevent up to  $t$  inside attackers, who are legitimate users, colluded together to forge tokens.

**Basic one-time group authentication protocol**

From now on, we assume that there are  $j$  users with their tokens  $t < j \leq n$ , who want to execute the group authentication protocol.

$f(x_1), f(x_2), \dots, f(x_j)$  where

The basic idea of this protocol is that each user releases the token obtained from the GM during registration. If all released tokens are valid, the interpolation of the released tokens can reconstruct the secret  $s$ . The published one-way hash of the secret is used to compare with the one-way hash of the reconstructed secret.

**Theorem 1.** *Protocol 1 can detect any number of illegitimate users.*

*Proof.* If there is illegitimate user who does not own a valid token on the polynomial

$f(x)$ , the

reconstructed secret will be different from the secret  $s$ . Thus, Protocol 1 can detect any number of illegitimate users. □

**Protocol 1: One-time group authentication protocol**

- **Step 1.** Each user  $M_i$  reveals his token  $f(x_i)$ , to all other users simultaneously.
- **Step 2.** After knowing all tokens,  $f(x_i)$ , for  $i = 1, 2, \dots, j$ , following Lagrange interpolating formula, each user computes  $s' = \sum_{i=1}^j f(x_i) \frac{-x_r}{x_i - x_r} \pmod{p}$ . If  $H(s) = H(s')$ , all users have been authenticated successfully; otherwise, there are illegitimate users.

*Remark 2.* This is a one-time authentication protocol since the secret and tokens are revealed to all users in this protocol. The authentication is no longer a one-to-one authentication and it is a many-to-many authentication. The proposed protocol is very efficient to authenticate multiple users belonging to the same group without revealing identity of each user.

### 3. GROUP AUTHENTICATION PROTOCOL WITHOUT REVEALING TOKENS

In Protocol 1, since tokens are revealed to all users, each token can only be used for one-time authentication. In addition, the secret  $s$  is also exposed to users in Protocol 1. In the following discussion, we propose a way to protect tokens. In addition, the secret does not need to be recovered in each authentication. Our authentication is based on the property of strong  $t$ -consistency in Section 2.2.

#### Group authentication protocol without revealing tokens

In the following protocol, it can be achieved authentication without revealing tokens and the secret. The basic idea of our approach uses the property of strong  $t$ -consistency. Let each user select a random polynomial with  $(t-1)$ -th degree and generate shares for other users. Then, each user releases the additive sum of his own token obtained from the GM during the registration and sum of shares of polynomials generated by users. Due to the property of secret

**Protocol 2:** Group authentication protocol without revealing tokens

- **Step 1.** Each user  $M_i$  selects a random polynomial,  $f_i(x)$ , with  $(t - 1)$ -th degree. For the polynomial  $f_i(x)$ , user  $M_i$  computes shares as  $f_i(x_r)$ , for  $r = 1, 2, \dots, j, r \neq i$ , for other users. User  $M_i$  sends each share,  $f_i(x_r)$  to user  $M_r$  secretly.

- **Step 2.** After receiving  $f_r(x_i)$  for  $r = 1, 2, \dots, j$ , each user uses his token  $f(x_i)$  to compute

$$y_i'' = f_i(x) + \sum_{r=1}^j f_r(x_i) \pmod p$$

Each user releases his value  $y_i''$

- **Step 3.** After knowing  $y''$ , for  $i = 1, 2, \dots, j$ , each user checks whether they are strong  $t$ -consistent. If they are not strong  $t$ -consistent, there are illegitimate users; else, all users have been successfully authenticated belonging to the same group.

sharing homomorphism in Section 2.2, the released sums are shares of the secret polynomial  $f(x)$  of tokens and sum of polynomials generated by users. If all users act honestly and own valid tokens, the released sums should be strong  $t$ -consistent; otherwise, the released sums are not strong  $t$ -consistent. Since users do not need to reconstruct the secret in the protocol and the tokens have not been revealed directly, the dealer does not need to publish the one-way of the secret  $s$  during system set up and the tokens can be reused.

**Theorem 2.** Protocol 2 can detect any number of illegitimate users.

*Proof.* Due to the property of secret sharing homomorphism, each released value,

$y''$  in Step 2 is the share of additive sum of polynomials,

$$f(x) + \sum_{r=1}^j f_r(x)$$

$r = 1$

$\pmod p$ , with  $(t - 1)$ -th degree.

Thus, in Step 3, all released values,

$y''$ , for  $i = 1, 2, \dots, j$ , are strong  $t$ -consistent. If there is any illegitimate user who does not own a valid token,

$f(x_i)$ , the released values,

$y''$ , for  $i = 1, 2, \dots, j$ , are not strong  $t$ -consistent with very high probability.  $\Upsilon$

**Remark 3.** In Step 2, the token

$j$

$f(x_i)$

cannot be computed from the revealed value

$$y'' = f(x) + \sum_{i=1}^r f(x) \cdot i$$

$r=1$

(mod  $p$ ). Therefore, the tokens are protected unconditionally and can be reused for multiple authentications.

### Computational complexity

The most time-consuming operation for each user is to check the strong  $t$ -consistency of released values  $y_i$

" for

$i = 1, 2, \dots, j$ , in Step 3 of Protocol 2. Following our discussion presented in Section 2.2, checking strong  $t$ -consistency needs to compute the interpolating polynomial of values

$y''$ . The polynomial interpolation becomes the main computational task in our proposed protocol. However, the modulus  $p$  in our polynomial interpolation is much smaller than the modulus in most public-key cryptosystems, such as RSA cryptosystem [13]. In addition, not like conventional user authentication protocol that authenticates one user at a time, this proposed authentication protocol authenticates all users at once. Thus, the proposed authentication protocol is very efficient in comparing with all existing authentication protocols.

## 4. CONCLUSIONS

We propose a special type of group authentication which is specially designed for group communications such as the ad hoc wireless communication network. The proposed group authentication protocol is no longer a one-to-one type of user authentication and it is a many-to-many type of authentication that authenticates multiple users at once. We first propose a basic one-time group authentication protocol and then propose a general group authentication protocol without revealing tokens. Our proposed group authentication is very efficient since the computation is based on the computation of linear polynomial.

## REFERENCES

- [1] Benaloh J. C., (1987) Secret sharing homomorphisms: keeping shares of a secret, in: Proceedings of CRYPTO '86, LNCS 263, pp. 251-260.
- [2] Bruhadeshwar B. and Kulkarni S.S., (2011) Balancing revocation and storage trade-offs in secure group communication, IEEE Transactions on Dependable and Secure Computing, 8 (1): 58-73.
- [3] Catur Bhakti M. A., Abdullah A., and Jung L. T., (2007) EAP-based authentication for ad hoc network, in: Proc. 2007 Seminar Nasional Aplikasi Teknologi Informasi – SNATI'07, pp. C-133-



C- 137.

- [4] Caballero-Gil P. and Hernandez-Goya C., (2009) Self-organized authentication in Mobile ad-hoc networks, *Journal of Communications and Networks*, 11(5): 509-517.
- [5] Capkun S., Buttyan, L. and Hubaux J. P., (2003) Self-organized public-key management for mobile ad hoc networks, *IEEE Transactions on mobile computing*, 2(1):52-64.
- [6] Das M. L., (2009) Two-factor user authentication in wireless sensor networks, *IEEE Transactions on Wireless Communications*, 8 (3): 1086-1090.
- [7] Downard I., (2002) Public-key cryptography extensions into Kerberos, *IEEE Potentials*, 21(5): 30- 34.
- [8] Harn L. and Lin C., (2010) Strong verifiable secret sharing scheme, *Information Sciences*, 180(16): 3059-3064.
- [9] Ku W. C., (2005) Weaknesses and drawbacks of a password authentication scheme using neural networks for multiserver architecture, *IEEE Transactions on Neural Networks*, 16(4), 1002-1005.
- [10] Oppliger R., Hauser R., and Basin D., (2008) SSL/TLS session-aware user authentication, *Computer*, 41(3): 59-65.
- [11] Pirzada A. A. and McDonald C., (2004) Kerberos assisted authentication in mobile ad-hoc networks, in: *Proceedings of the 27th Australasian Computer Science Conference –ACSC’04*, 26(1), pp. 41-46.
- [12] Ren K., Yu S., Lou W., and Zhang Y., (2009) Multi-user broadcast authentication in wireless sensor networks, *IEEE Transactions on Vehicular Technology*, 58(8): 4554-4564.
- [13] Rivest R., Shamir A., and Adleman L., (1978) A method for obtaining digital signatures and public- key cryptosystems, *Communications of the ACM*, 21 (2): 120-126.
- [14] Sakarindr P. and Ansari N., (2010) Survey of security services on group communications, *IET Information. Security*. 4(4): 258-272.
- [15] Shamir A., (1979) How to share a secret, *Communications of the ACM*, 22(11): 612-613.
- [16] Yan J., Blackwell A., Anderson R., and Grant A., (2004) Password memorability and security: Empirical results, *IEEE Security & Privacy Magazine*, 2(5):25-31.