The Second Indonesian-Japanese Conference on
Knowledge Creation and Intelligent Computing (KCIC) 2013 *ISBN : 978-602-9494-68-6*

A Light-Weight Group Signature Scheme for Wireless Networks Based-on BBS Short Group Signature

Amang Sudarsono and Mike Yuliana
Division of Telecommunication Engineering, Dept. of Electrical Engineering,
Electronic Engineering Polytechnic Institute of Surabaya (EEPIS), Surabaya, Indonesia.
EEPIS Campus, Jalan Raya ITS Sukolilo, Surabaya 60111
Tel: +62(31) 594 7280; Fax: +62(31)594 6114
Email :amang@eepis-its.edu, mieke@eepis-its.edu

Abstract

In the natural context of wireless network environment, the communications between wireless nodes are more easily observed for the goal of the network traffic analysis. Thus, to enable a secure and anonymous communication system from thwarting of such analysis attacks would be strongly desirable. In this paper, we propose a secure and anonymous communication system using pairing-based group signatures. The achievement of secure and anonymous communication is performed by allowing all valid member wireless nodes of a particular privilege group to authenticate each other without revealing their own identities.

Keywords: group signature, anonymity, signer, verifier, wireless networks, authentication.

1. Introduction

Recently, there are numerous ubiquitous services growing rapidly along with the advancement of personal computer, laptop, smart phone, and other embedded devices. Almost services require an authentication or identification for accessing control and authorization. As the result the accessed services by a user can be linked and tracked, hence the system obtained the user preference access history. One of user-privacy problem solving in the privacy-preserving authentication systems is the use of group signature, since it is very practical and able to provide not only the anonymity, but also unlinkability and untraceability.

Group signature is a kind of digital signature based on public key (one is given to a privilege group, not to one user). The group signature was introduced by Chaum and Heyst [13] for the first time. Currently, group signature also becomes one of the main topics in the cryptographic technology and many researchers actively have been taking in account such topic of interest [2–9]. The group signature scheme allows the users to sign messages without revealing their own privacy information (i.e. identity). In case of misuses or other reasons, there is an authority called group manager (GM) can trace the signer. Many applications of group signaturealso have been proposed and studied [6, 9]. In this paper, we consider the use of group signature for communication protocol in the wireless mobile networks such that able to provide a secure and anonymous communication.

Again, in the current era of pervasive computing, where ubiquitous services exist as an integrated part of our environment settings. Thus, computers, handhelds, gadgets, and other mobile devices are going to beexchanging messages nodes with each other (e.g., wireless networks, sensor networks, vehicle-2-vehicle communications [6, 9–11]). To satisfy these systems requirement such that they are able to work properly, every messagehas todeliverthe most important information of authentication. However,the system requirements on the authentication are depend on any cryptographic solution. Ideally, such requirements should fulfill the following matters simultaneously:

a. Low bandwidth consumption: that due to the limited spectrum available for wireless communication, sensor network, and vehicular communication. Thus, a mechanism to achieve any shorter than RSA signatures is needed (i.e., shorter signature size, shorter processing time, shorter bit-length, lower power consumption).
b. Fast verification for large numbers of messages from different sources: that due to the suggestion of [12]whereas the safety message re-transmission of vehicles is done every 300ms to all other vehicles within 110 meters of a minimum range. This means that it is much more critical in the authentication phase. Therefore, it is better if the verification process is faster than generation process.
c. Privacy-friendly or anonymity: that due to users-privacy information should be protected from the information involved for every authentication process.

One of applications requiring group signatures in wireless network implementation for IEEE802.1X-based wireless protocol [6] showed the effectiveness of using group signature to achieve a user-privacy enhancing authentication. The modification of verifier-local

revocation group signature [7] such that it is to be easily and efficiently adopted and applied for anonymous authentication in the IEEE802.1X-based wireless communication protocol. However, due to the group signature scheme considered the user revocation function, it suffered from the number of revoked users. Another application of using group signature is the application for cloud computing environment [9] which involves mobile devices, sensor networks, embedded systems, etc. This work investigatedthe differences between group signatures and ECC for client devices and servers in cloud computing technology by introducing a modified existing High-Level Synthesis in order can be adapted with group signature implementation into FPGA board hardware.

In this paper, we propose a light-weight and simple scheme of group signature from short group signature scheme (BBS signature) [2] in order to be easily and efficiently applied for wireless mobile networks, sensor networks, and other ubiquitous devices communication protocol.

2. Previous Works

Digital signatureshave overhead computation time.Thus, researchers have tried to find out alternative protocols designed to suitable signatures over many packets. The desirable signatures require verifiers to acquire many packets before verifying. Other approaches including the shortsignatures are inappropriate for the spread-settings, because verification process requires interaction with the signer frequently.

In 2001, BLS signature [3] was developed. The scheme is based on a pairing-based group signature in 170-bit length that provides the same security level with 1024-bit RSA. Thediscovery was followed by many researchers to achieve more efficient signature schemes. As the result of advancement, many signature variants have been proposed, some of them are privacy-friendly and shorter group signatures [2, 4, 5, 7, 8].

BBS signatures [2] is a short group signature scheme based on a non-interactive zero-knowledge protocol for Strong Diffie-Hellman (SDH) and Decision Linier assumptions in the bilinear pairing groups. The scheme employees a bilinear map,$e: G_1 \times G_2 \rightarrow G_T$. The group G_1 has a short representation and the length of group signature is under 200 bytes. Meanwhile, the advantages of this scheme are the signature generation that requires no bilinear pairing computation, the verification requires a single pairing, and both signature generation and verification need a few exponentiations. Therefore, based on BBS signature, we consider adopting the scheme for our secure and anonymous wireless network communication protocol and its implementation.

3. Bilinear Groups and Complexity Assumption

Firstly, we describe the concept related to bilinear maps. The bilinear map notation can be defined as follows:

a. G_1 and G_2 are two multiplicative cyclic groups of prime order p.
b. $g_1$is a generator of G_1 and g_2 is a generator of G_2.
c. φ is a computable isomorphism from $G_2$to G_1 by the isomorphism function $\varphi(g_2) = g_1$; and
d. e is a computable map, $e: G_1 \times G_2 \rightarrow G_T$ with the following properties:
   - Bilinearity: for all $u \in G_1, v \in G_2$ and $a, b \in Z$, where$e(u^a, v^b) = e(u, v)^{ab}$.
   - Non-degeneracy: $e(g_1, g_2) \neq 1$.

Secondly, we use the following assumptions for the security requirements.

a. Strong Diffie-Hellman Assumption.
   Let G_1 and G_2 be cyclic groups of prime order p. There is possibility that $G_1 = G_2$. Let$g_1$be a generator of $G_1$and g_2 is a generator of G_2.

   q-StrongDiffie-Hellman Problem (q-SDH): The q-SDH problem in (G_1,G_2) is defined as follows:
   Given a (q+2)-tuple $(g_1, g_2, g_2^{\gamma}, g_2^{(\gamma^2)}, \ldots, g_2^{(\gamma^q)})$ as input, and the output is a pair $(g_1^{(1/(\gamma+x))}, x)$, where $x \in Z_p^*$. An algorithm Ahas advantage $\in$ in solving q-SDH in (G_1,G_2)if

$$P_r\left[A\left(g_1, g_2, g_2^{\gamma}, \ldots, g_2^{(\gamma^q)}\right) = \left(g_1^{\frac{1}{\gamma+x}}, x\right)\right] \geq \in.$$

   Where the probability is over the random choice of generator g_2 in G_2 of γinZ_p^*, and of the random bits of A.

b. Decision Linear Diffie-Hellman Assumption.
   By using g_1 in G_1 as above, along with arbitrary generators u, v, and $h$of G_1, consider the following:

   Decision Linear Problem in G_1: Given u, v, u^a, v^b, h^c in G_1 as input, the output is**yes**if $a + b = c$ or**no** otherwise.
   More precisely, the definition of the advantage algorithm$A$ in deciding the Decision Linear problem in $G_1$is as:

$$\text{Adv Linear}_A = \left| P_r\begin{bmatrix} A(u, v, h, u^a, v^b, h^{a+b}) = yes \\ : u, v, h \leftarrow G_1, a, b \leftarrow Z_p \end{bmatrix} - P_r\begin{bmatrix} A(u, v, h, u^a, v^b, \tau) = yes \\ : u, v, \tau \leftarrow G_1, a, b \leftarrow Z_p \end{bmatrix} \right|$$

   The probability is over the uniform random choice of the parameters to A, and over the coin tosses of A.

4. Light-Weight Group Signature Scheme for Wireless Networks

Firstly, we review the security requirements of communication protocol. Secondly, we review the BBS short group signature as our adoption scheme in the proposed communication protocol. Finally, in details we describe the four phases: KeyGeneration (KeyGen) phase, Registration phase, Authentication phase, and Tracing phase in our proposed communication protocol. The first phase is for generating public and secret parameters. The second phase is the user or wireless nodesregistration to the group manager authority andobtaining some secret information used for authentication process. The third phase is for wireless node authentication to each other. One node acts as the signer and another will act as the verifier. The signer generates his group signature and the verifier verifies the signer's signature to prove that the signer is a legitimate user without revealing any privacy information of signer. The fourth phase is for the GM to trace the user about his history records.

4.1 Security Requirements

Some security requirements of secure and anonymous communication protocol are listed as follows.

User anonymity: No one can identify the user or wireless nodes.

Unlinkability: There are two or more signatures, no one can distinguish whether these signatures are related or not.

Untraceability: No one can trace user's records. The goal is protecting user's privacy, which means that the identity and related secret information of the user cannot be revealed.

Unforgeability: no one except users of the group is able to generate a valid signature.

Confidentiality: Only GM can obtain user's communication history through the Tracing mechanism.

Integrity: No one can modify the message content.

Authentication: The user can request services to other users or gateway for confirming the legitimacy of the user.

4.2 BBS Short Group Signature

Boneh et al. [2] proposed a short group signature scheme to hide signer's identity in the signature by using linear encryption based on decisional linear assumption. The total signature length is 1533 bits or 192 bytes for the same security level with 1024-bit RSA. In this scheme, there are 3 players who involved in the system (see Fig. 1):

a. Group Manager: it has an authority to issue the key (group public key $\mathsf{gpk} = g_1, g_2, h, u, v, w)$ and group secret key $\mathsf{gmsk} = (\vartheta_1, \vartheta_2, \gamma)$, and also user's private key pair) through a Setup algorithm and open signer's identity through an Open algorithm.
b. User or signer: the entity who joins the group. He signs a message to prove himself as a legitimate user anonymously using his private key $\mathsf{gsk}[i] = (A_i, x_i)$ issued by GM through Sign algorithm.
c. Verifier: the entity who verifies user's signature to check whether the user is valid user or not anonymously through Verify algorithm.

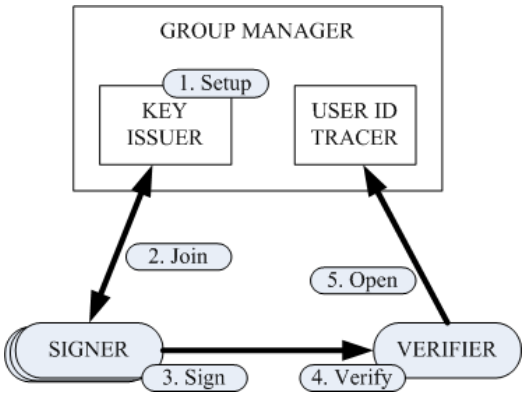

Fig 1. Involved players and procedures in short group signature.

The detail procedure of this scheme is as follows:

a. Setup: the GM selects secret key $\mathsf{gmsk} = (\vartheta_1, \vartheta_2, \gamma)$ and group public key $\mathsf{gpk} = (g_1, g_2, h, u, v, w)$, where $u^{\vartheta_1} = v^{\vartheta_2} = h$ and $\mathsf{w} = h^{\gamma}$. While g_1 and g_2 are generators of the bilinear groups $G_1$and G_2.
b. Join: user selects his secret key $x_i \in_R Z_p^*$ randomly. Then, GM computes $A_i = {g_1}^{1/(\gamma + x_i)}$ and sends A_i to user. Here, the user has his secret key $\mathsf{gsk}[i] = (A_i, x_i)$.
c. Sign : user generates a group signature $\sigma = (T_1, T_2, T_3, \mathsf{c}, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$ for the message M by suing his secret key.
$T_1 = u^{\alpha}, T_2 = v^{\beta}, T_3 = A_i h^{\alpha+\beta}$. Here, $T_1, T_2,$and T_3 are linear encryption results for blinding $A_i, \alpha$and β. Then, the user computes
$R_1 \leftarrow u^{r_\alpha}, R_2 \leftarrow v^{r_\beta}, R_3 \leftarrow e(T_3, g_2)^{r_x}.\, e(h, \mathsf{w})^{-r_\alpha - r_\beta}.\, e(h, g_2)^{-r_{\delta_1} - r_{\delta_2}}, R_4 \leftarrow T_1^{r_x}.\, u^{-r_{\delta_1}}, R_5 \leftarrow T_2^{r_x}.\, v^{-r_{\delta_2}}$
with random blinding values $r_\alpha, r_\beta, r_x, r_{\delta_1}, r_{\delta_2}$. Also he computes a challenge $c = Hash(M, T_1, T_2, T_3, R_1, \ldots, R_5)$ using random numbers $s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2}$ which are the values for zero-knowledge proof of (A_i, x_i).
d. Verify: the verifier verifies the user's signature on given message M and signature $\sigma = (T_1, T_2, T_3, \mathsf{c}, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$, the verifier computes $\tilde{R}_1, \ldots, \tilde{R}_5$ such that $\tilde{R}_1 \leftarrow u^{s_\alpha}.\, T_1^{-c}, \tilde{R}_2 \leftarrow v^{s_\beta}.\, T_2^{-c}, \tilde{R}_4 \leftarrow u^{-s_{\delta_1}}.\, T_1^{s_x}$,

$\tilde{R}_5 \leftarrow v^{-s_{\delta_1}}.T_2^{s_x}$, and $\tilde{R}_3 \leftarrow e(T_3,g_2)^{s_x}.e(h,\mathsf{w})^{-s_\alpha - s_\beta}.\mathsf{e}(h,g_2)^{-s_{\delta_1}-s_{\delta_2}}.(e(T_3,w)/e(g_1,g_2))^{c}$.

Then, the verifier checks if c is equal to $c' = Hash(\mathsf{M}, T_1, T_2, T_3, \tilde{R}_1, \ldots \tilde{R}_5)$ or not. The verification is successfulif $c = c'$.

e. Open: on given message M and signature $\sigma = (T_1, T_2, T_3, \mathsf{c}, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$, the GM checks the validity of signature and opens the signer's secret A_i as $A_i = T_3/(T_1^{\vartheta_1}.T_2^{\vartheta_2})$, if the signature is valid, then A_i is a part of signer as the signer identity.

4.3 The Proposed Protocol

The proposed protocol is based on the BBS short group signature [2] and the hash function technology. The detail procedureis described in Fig. 2.

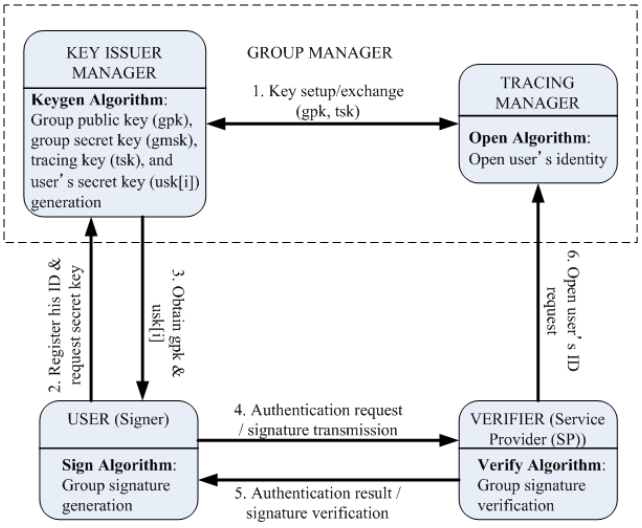

Fig 2.Proposed communication protocol.

We separated the authority of GM into two entities, Key Issuer Manager and Tracing Manager. Key Issuer Manager has the authority to issue group public key (gpk), group secret key (gmsk), tracing key (tsk), and user's secret key usk[i] for the successful joining users. While the Tracing Manager has the authority to trace and open the user's identity information from the user's signature obtained from the verifier (SP) who requested opening user's identity, in case of misuses activities, contract expiration date of services, or other reasons. We define our protocol procedures into 4 phases which are described in detail as follows:

Phase 1: KeyGen Algorithm.

This is the randomized algorithm with the input parameter is$n$, the number of users of a privilege group. Then, the Key Issuer Managerproceeds the following steps:

a. Select a generator $g_2 \in_R G_2$uniformly at random. Set $g_1 \leftarrow \varphi(g_2)$. Select $h \in_R G_1, \vartheta_1, \vartheta_2 \in_R Z_p^*$ and set $u, v \in G_1$ such that $u^{\vartheta_1} = v^{\vartheta_2} = h$.
b. Select $\gamma \in_R Z_p^*$ and set $w = g_2^{\gamma}$.
c. Select $g \in_R G_1$and $\mathsf{s} \in_R Z_p^*$. Then, compute$S = g^s$.
d. Output the group public key $\mathsf{gpk} = (g, g_1, g_2, h, u, v, w, S, p, G_1, G_2, e)$, group secret key $\mathsf{gmsk} = (\vartheta_1, \vartheta_2, \gamma)$, and the tracing secret key $\mathsf{tsk} = (s)$.
e. Distribute gpk and tsk to Tracing Manager.

Phase 2: Registration Protocol.

This is a communication protocol between Key Issuer Manager and a joining user. The i-th user joins to the group by processing the following steps:

a. User i selects $x_i, z_i' \in_R Z_p^*$, and computes $H_i = u^{x_i}v^{z_i'}$ and $Q_i = g^{x_i}$. Where the user's ID is embedded into his secret key x_i.
b. User i sends Key Issuer Manager($H_i$and Q_i), and proves that $H_i = u^{x_i}v^{z_i'}$ and $Q_i = g^{x_i}$ by a signature proof of knowledge(SPK). Where SPK is performed by utilizing Fiat-Shamir heuristic [14] conversion signatures using a hash function from zero-knowledge proof of knowledge (PK) as well as in [2– 9], where a signer can convince a verifier of knowledge by relation on representations. We call such mechanism as signature PK's or SPK.
c. Key Issuer Manager selects $y_i, z_i'' \in_R Z_p^*$, and computes $A_i = (g_1 H_i v^{z_i''})^{1/(\gamma + y_i)}$. Then the Key Issuer Manager sends (A_i, y_i, z_i'') to user i. Key Issuer Manager adds (i, y_i, Q_i) to his Group List (GL),which is the database of users in the group.
d. Upon receiving (A_i, y_i, z_i''), user$i$ computes $z_i = z_i' + z_i''$, and outputs $\mathsf{usk}[i] = (A_i, x_i, y_i, z_i)$.

Phase 3: Authentication Protocol.

This is an authentication protocol between the signer and the verifier. This protocol comprises into two algorithms, Sign algorithm and Verify algorithm. Detail descriptions of algorithms are described as follows:

Signature generation: Sign Algorithm.

This algorithm is performed by the user (signer) to authenticate himself to a verifier who acts as an SP for accessing services offered by verifier, where the inputs of signing algorithm are the group public key gpk, the signer secret key usk[i], and a signed message,$M \in \{0,1\}^*$. The algorithm is performed as follows:

a. Select $\alpha, \beta \in_R Z_p^*$ and set $\mu = -z_i - \alpha y_i$. Then, compute $T_1 = A_i v^{\alpha}$and $T_2 = u^{\beta}$.
b. Select $f \in_R Z_p^*$. Compute $\tilde{S} = S^f$and $\tilde{T} = g^{x_i + f}$.

c. The SPK X is computed as follows:

$$X = \mathrm{SPK}\{(x_i, y_i, \alpha, \beta, \mu, f): \frac{e(g_1, g_2)}{e(T_1, w)} = e(T_1, g_2)^{y_i} e(u, g_2)^{-x_i} e(v, g_2)^{\mu} e(v, w)^{-\alpha}, T_2 = u^{\beta}, \tilde{S} = S^{f}, \tilde{T} = g^{x_i + f}\}(M).$$

d. Pick blinding factors: $r_{x_i}, r_{y_i}, r_\alpha, r_\beta, r_\mu, r_f \in_R Z_p^*$.

e. Compute:
$R_1 = e(T_1, g_2)^{r_{y_i}} e(u, g_2)^{-r_{x_i}} e(v, g_2)^{r_\mu} e(v, w)^{-r_\alpha}$,
$R_2 = u^{r_\beta}$, $R_3 = S^{r_f}$, $R_4 = g^{r_{x_i} + r_f}$.

f. Compute a challenge $c \in_R Z_p^*$ as:
$c = Hash(\mathsf{gpk}, M, T_1, T_2, \tilde{S}, \tilde{T}, R_1, R_2, R_3, R_4)$.

g. Compute responses:
$s_{x_i} = r_{x_i} + cx_i$, $s_{y_i} = r_{y_i} + cy_i$, $s_\alpha = r_\alpha + c\alpha$, $s_\beta = r_\beta + c\beta, s_\mu = r_\mu + c\mu, s_f = r_f + cf \in Z_p^*$.

h. Output the group signature:
$\sigma = (T_1, T_2, \tilde{S}, \tilde{T}, c, s_{x_i}, s_{y_i}, s_\alpha, s_\beta, s_\mu, s_f)$.

Signature verification: Verify Algorithm.

This algorithm is performed by verifier with the inputs are gpk, a target signature σ, and the message $M \in \{0,1\}^*$. The signatureσ is verified as follows:
Signature check: check whether σ is valid or not by using SPK X as follows:

a. Re-derive $\tilde{R}_1, \tilde{R}_2, \tilde{R}_3, \tilde{R}_4$as:

$$\tilde{R}_1 = e(T_1, g_2)^{s_{y_i}} e(u, g_2)^{-s_{x_i}} e(v, g_2)^{s_\mu} e(v, w)^{-s_\alpha} \cdot \left(\frac{e(g_1, g_2)}{e(T_1, w)}\right)^{-c},$$

$$\tilde{R}_2 = \frac{u^{s_\beta}}{T_2^c}, \tilde{R}_3 = \frac{S^{s_f}}{\tilde{S}^c}, \tilde{R}_4 = \frac{g^{s_{x_i} + s_f}}{\tilde{T}^c}.$$

b. Re-derive the challenge$c' \in_R Z_p^*$ as:
$c' = Hash(\mathsf{gpk}, M, T_1, T_2, \tilde{S}, \tilde{T}, \tilde{R}_1, \tilde{R}_2, \tilde{R}_3, \tilde{R}_4)$.
If $c = c'$,the signature is valid, otherwise signature is invalid.

Phase 4: Tracing

The input of this algorithm are gpk, the traced signatureσ, message M, and the tracing secret key tsk. The Tracing Manager traces and identifies the signer as follows:

a. Verify the traced signature by using the above Verify algorithm.
b. If the signature is valid, compute $Q_i = \tilde{T} / \tilde{S}^{-s}$, using the tracing key $\mathsf{tsk} = (s)$.
c. Output i.

4.4Efficiency and Security Consideration

To confirm the better efficiency of the proposed scheme, we give the efficiency comparison of Sign algorithm and Verify algorithm, excluding Join-related and Open-related parts. Table 1 shows the comparison of the computation costs. As the overhead, the proposed scheme needs slightly more pairing computation and exponentiation on pairing computation. On the other hand, the exponentiation on G_1 is smaller than [2]. However, the computation cost on G_T and pairing are more expensive than the computation cost on G_1. This comparison result means that the proposed scheme is slightly more overhead than in [2], since the number of G_T and pairing computations is higher.In the Sign algorithm of proposed scheme, the number of G_T computation is 4 and pairing computation is also 4, while in the previous scheme [2] only consumes 3 G_T computation and 3 pairing computation. Meanwhile, the Verify algorithm in the proposed scheme consumes 6 G_T computation and 5 pairing computation, whereas in the previous scheme [2] consumes 5 G_T computation and 4 pairing computation (see Table 1).In this case, the overhead of each process of proposed scheme comprises of a G_T computation and a pairing computation comparing with previous scheme [2]. However, in our proposed scheme has an advantage to be easier implemented for common authentication system, since we providedthe additional secret components, (y_i, z_i, Q_i), which are used along with the main secret component of user x_i.Secret key z_i is formed from user secret key z_i' when the user registering himself to the Key Issuer Manager which is embedded in his user secret key part $H_i = u^{x_i} v^{z_i'}$. Upon proving the validity of $H_i = u^{x_i} v^{z_i'}$, GM embeds his secret key z_i'' into the user secret key component $A_i = (g_1 H_i v^{z_i''})^{1/(\gamma + y_i)}$ along with his own secret key y_i. Hence, the part of user secret key, $A_i = (g_1 u^{x_i} v^{z_i'} v^{z_i''})^{1/(\gamma + y_i)} = (g_1 u^{x_i} v^{z_i})^{1/(\gamma + y_i)}$, where $z_i = z_i' + z_i''$. In addition, this mechanism provides more secure authentication process than previous scheme [2].

Table 1. Comparison of computation costs of Sign and Verify algorithm.

Scheme	Computation cost of Sign algorithm Computation cost of Verify algorithm
[2]	$9E(G_1) + 3G_T + 3E(G_T)$ $8E(G_1) + 5G_T + 4E(G_T)$
Proposed scheme	$7E(G_1) + 4G_T + 4E(G_T)$ $6E(G_1) + 6G_T + 5E(G_T)$

Note:
$E(G_1)$: computation of exponentiation on G_1 component.
G_T: computation on pairing.
$E(G_T)$: computation of exponentiation on pairing.

5. Performance Measurement

In this section, we present the experiment results to show the efficiency of the proposed scheme.

We measured the performance of our proposed scheme in a desktop PC. The specification of PC is shown in Table 2. Table 3 shows the time comparison of signing, verification and opening algorithm between previous scheme [2] and our proposed scheme.

Table 2. Specification of H/W used in experiment.

Specification of	Remarks
Software	gcc-4.4, gmp-5.1.0, pbc-lib-0.5.12
O/S	Ubuntu Linux kernel-2.6.35
CPU	Intel Core i5 3.20GHz
RAM	2GB

Overall the total authentication time of our proposed scheme takes about 80 ms, while in previous scheme [2] is only about 60 ms.Table 1shows thatthe process ofsigningandverificationhas a difference of one computation on G_T, whereas computationaltimeon G_T is about 7ms.So the totaltimedifference of G_T and the pairing preparationis only about 15ms.

The Opening algorithm takes the highest cost, since the Opening algorithm consists of signature check (verification) and the computation of tracing the user's identity (see Section 4.2 and 4.3).

Table 3.Comparison of computation time of Keygen, Sign, Verify and Open algorithm.

Time	Scheme [2] (ms)	Proposed scheme (ms)
Keygen	119.13	137.18
Signing	21.52	35.24
Verification	36.92	49.96
Opening	74.84	79.67

6. Conclusion

We have presented a light-weight group signature scheme for the implementation of secure and anonymous communication protocols in the wireless networks which is suitable for mobile devices, wireless sensor network devices, or other embedded system devices. Our proposed group signature was constructed from the BBS short signature scheme withthe total time of authentication is only 80ms in the current condition of desktop PC.

7. Future Works

Our future works include the implementation of the proposed scheme into wireless mobile devices and its investigation, the implementation of secure and anonymous communication protocol on the online transaction scenarios, and the further improvement of group signature scheme.

References

[1] D. Boneh and X. Boyen, "Short signatures without random oracles and the SDH assumption in bilinear groups". Journal of Cryptology, 21(2): pp. 149–177, 2008.

[2] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures". In CRYPTO '04, Vol 3152 of LNCS, pp. 45–55, 2004.

[3] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing". Journal of Cryptology, 17(4): pp. 297–319, 2004.

[4] D. Boneh and H. Shacham, "Group signatures with verifier-local revocation". In CCS, pp. 168–177, 2004.

[5] A.L. Ferrara, M. Green, S. Hohenberger, M.O. Pedersen, "Practical Short Signature Batch Verification". In http://eprint.iacr.org/2008/015.pdf, pp. 1–24, January 21, 2009.

[6] A. Sudarsono, T. Nakanishi, Y. Nogami, and N. Funabiki, "Anonymous IEEE802.1X authentication system using group signatures".Journal of Information Processing, Vol. 18, pp. 63–76, March, 2010.

[7] T. Nakanishi and N.Funabiki, "A short verifier-local revocation group signature scheme with backward unlinkability".In 1st International Workshop on Security (IWSEC 2006), LNCS 4266, Springer Verlag, pp.17–32, October 2006.

[8] J. Camenisch, S.Hohenberger, and M.O. Pedersen, "Batch verification of short signatures". In EUROCRYPT '07, Vol 4515 of LNCS, pp. 246–263. Springer, 2007.Full version at http://eprint.iacr.org/2007/172.

[9] S. Morioka, J. Furukawa, Y. Nakamura, K. Sako, "Architecture optimization of group signature circuits for cloud computing environment". In The 17th Workshop on Synthesis And System Integration of Mixed Information Technologies, pp. 497–502, March, 2012.

[10] SeVeCom. Security on the road.http://www.sevecom.org.

[11] Car 2 Car. Communication consortium.http://car-to-car.org.

[12] M. Raya and J.P.Hubaux, "Securing vehicular ad hoc networks".Journal of Computer Security,pp. 39–68, 2007.

[13] D. Chaum and E. van Heyst, "Group signatures". In D.W Davies, editor, Proceeding of Eurocrypt 1991, Vol 547 of LNCS, pp. 257–265, April 1991.

[14] A. Fiat and A. Shamir,"How to prove yourself: Practical solutions to identification andsignature problems". In A. M. Odlyzko, editor, Proceedings of Crypto 1986, volume 263 ofLNCS, pp 186–194. Springer-Verlag, Aug. 1986.