Tatra Mt. Math. Publ. **37** (2007), 23–33

Tatra Mountains
Mathematical Publications

PUBLIC GROUP KEY CRYPTOGRAPHY

Sigurd Eskeland

ABSTRACT. Several conference key establishment protocols have in the recent years been proposed to provide secure group communication. In this paper, we introduce the concept of public group key cryptography as an extension to regular conference key agreement, and propose an efficient and practical public group key cryptosystem that enables participants of groups, without an online trusted third party, to flexibly establish public/private key pairs that represent the groups. By means of private group keys, secure communication is provided internally between the group members, and moreover, outsiders can by means of the public group key address the group confidentially. A multi-party signature provides certification of the public group key towards the group participants.

1. Introduction

With the evolution of Internet and wireless networks and with increasing focus on group collaboration, there has been a correspondingly increased focus on secure group communication, i.e., two or more participants that are physically located on different locations can confidentially establish on a secret key over an insecure public network. Key establishment protocols can be categorized into two concepts: 1) *Key agreement* where each participant contributes equally to the value of the session key, and 2) *key distribution* (or key transfer) where one a trusted party or participant generates a secret session key and transfers it confidentially and securely to the legitimate participants. Key agreement protocols involving more than two participants are called conference key agreement or multi-party key agreement protocols.

There may be situations when it is desirable that outsiders can address a group securely and confidentially. A naive approach for this in the context of secure group key establishment protocols could be to include the outsider into the group, and then re-run the protocol in order to re-establish a new secret group

2000 Mathematics Subject Classification: 94A60, 11T71.
Keywords: public group key cryptography.

key. The confidentiality of the secret information shared within the group would accordingly be compromised, since the new participant would be sharing the key with the group and has therefore access to subsequently exchanged confidential information that is only to be disclosed to the original members.

A proper solution could be to incorporate a public key that would publicly represent the group where the complementary secret key would consequently be known to the group members only. This way, the following is achieved: 1) The group members can now communicate securely among themselves, and 2) outsiders can address the group securely by means of the public key.

Centralized solutions involving online trusted third parties (TTP) providing key generation and secure key distribution, may conform poorly to distributed and wireless environments. By extending the concept of conference key agreement, we propose a public group key cryptosystem, where group members can collaboratively establish corresponding public/private group keys pairs over a wireless network without involving an online TTP. In order to make the public keys trustable, some kind of certification is required. The proposed cryptosystem provides identity-based signatures that certify public group keys toward identities of the originating users and time of establishment.

1.1. Related work

Threshold-oriented cryptosystems is class of group-oriented cryptosystems [1–5] that are based on the concept of public key cryptography where a group, consisting of a numbers, is represented by a public key. Although our scheme provides a public key representation of groups, it differs fundamentally from those due to that it is not a threshold scheme, and that it is identity-based. We will nevertheless discuss a few such schemes since they partially relates to our scheme.

The first threshold-oriented cryptographic protocol was proposed in [2]. In this scheme, there is one public key and a number of corresponding secret shares that together represent the corresponding secret key. The public key represents the group, and the public key and secret shares are generated by a trusted third party (TTP). Each participant is handed a secret share by which each participant performs a partial decryption of the ciphertext. A minimum number of participants, defined by the threshold, must combine their partial decryptions in cooperation with a trusted party to performs the combiner function to reconstruct the plaintext. Due to the (unsigned) public key, the scheme provides anonymity, keeping the members of the group anonymous to outsiders.

The protocol is presented in [4] and subsequently in [5] is an identity-based threshold-decryption scheme based on self-certifying public keys [13]. The participants compute the plaintext themselves from their partial decryptions, thereby eliminating an online trusted party for this. However, this scheme requires the sender to encrypt the plaintext for each of the members of the corresponding group that he or she wants to confidentially address. In contrast to encrypting once by means of one public key, this scheme imposes an unreasonably high overhead considering computation and bandwidth.

2. Public group key

By means of a public group key, outsiders can confidentially address groups and the group members can communicate secretly among themselves. The following list indicates security requirements for the proposed key establishment protocol:

- The protocol must resist passive attacks (i.e., eavesdropping), so that secret group keys remain confidential to the concerning parties only.
- The protocol must resist active attacks so that attempts of substitution of and alternation of key establishment messages will be detected.
- The establishment of group keys is to be performed in a distributed and contributory fashion so that each participant contributes equally to the key, thereby providing unique and "fresh" keys for each session.
- Forward secrecy is defined as if a long-term user key is compromised, session keys previously established with that long-term user key cannot be compromised [12, p. 50]. The protocol must provide forward secrecy.
- Public group keys must be certifiable offline (without an online trusted party) towards the identities of all the originating participants, and time of establishment.
- No online TTP or combiner functions.
- A nice but not really mandatory property is public reconstruction of the public group key, making the public group key privately verifiable towards the secret group key, thereby providing implicit key authentication.

An important property for a public key is to provide proof of context. In order for a public key to be *trustworthy* for the outside world or valid, outsiders must be certain that the owner of this public key is *really* the person he thinks it is. Otherwise, an unauthorized individual could cheat and obtain this confidential information of substituting public keys with his own. It is therefore essential that the association between public keys and their owners can be verified. In the context of our scheme, individuals running the protocol would consequently constitute and thereby define the groups that the established keys would represent. It would be likewise mostly preferably to that the association between public group keys and the identities of the group members they represent, and even the time of establishment, could be verified. The issue of statically linking public keys with identities is addressed by certificates and in [13] by the concept of self-certified public keys. This does not comply with group key agreement, where keys are flexibly established on-the-fly by a number of participants, independently of a TTP. Conforming to distributed key agreement in absence of a TTP, the proposed protocol employs a homomorphic identity-based signature scheme for establishment of multi-party signatures verifying the corresponding public group keys.

3. A practical public group key cryptosystem

In this section, the public group key protocol is presented. In 1994, Burmester and Desmedt generalized the well-known Diffie-Hellman two-party key agreement protocol into a conference key agreement protocol [6]. In [7], this was moreover generalized into a multiplicative and an additive variant having the secret keys on the form respectively as $\mathcal{Z} = k_{1,2} \cdot k_{2,3} \cdot \ldots \cdot k_{m,1}$ and $\mathcal{Z} = k_{1,2} + k_{2,3} + \ldots + k_{m,1}$. Our protocol is based on the additive variant and is very efficient as it requires only two rounds (two broadcasts per participant) + a third round for establishment of the multi-party public group key signature. Due to broadcasting, it is well suitable for wireless environments. The protocol is identity-based due to that each participant is assigned identity-based long-term secret user keys.

3.1. Setup

A trusted center, a trusted third party (TTP) computes the product of two secret, large prime numbers $n = p \cdot q$. According to the RSA cryptosystem, the center selects a number e that is relative prime to $\phi(n)$, and computes a secret integer d so that $e \cdot d \equiv 1 \pmod{\phi(n)}$. The TTP publishes n and e.

The only purpose for the TTP is to provide long-term secret ID-based user keys. Long-term user keys are required for user computation of signatures certifying key agreement messages and public group keys. The TTP computes for each user U_i the long-term secret user keys, $s_i = ID_i^d \pmod{n}$ and $h_i = \alpha^{d \cdot f(ID_i)} \pmod{n}$ where f denotes a secure hash-function, d is only known by TTP, and α is a public primitive root of GF(p) and GF(q).

It is assumed that the user identities consist of meaningful information like names, addresses, etc., or that the identities are posted in tamper-free tables or secure bulletin boards for verification. Otherwise, adversaries can produce fake user keys s' representing *meaningless* identities id' according to $id' = {s'}^e \pmod{n}$.

3.2. Group key generation

The group $\mathcal{U} = \{U_1, \dots, U_m\}$, consisting of m members, is arranged according to a logical ring structure so that U_m and U_1 are an adjacent pair of users, and that $U_{m+i} = U_i$. Note that in general that the indexing is circular, so that if the index j is $j < 1$, then it corresponds to the index $i = j + m$, or if $j > m$, then the corresponding index is $i = j - m$, where $1 \leq i \leq m$. Thus, the indices j and $j + m$ are equivalent: $j \equiv j + m$. The protocol goes as follows:

Round 1. Each adjacent pair of users, U_i, U_{i+1}, establish a secret shared key by means of a secure two-party key agreement (TPKA) protocol. That is, every user U_i establishes with users U_{i+1} and U_{i-1} the two secret keys, $k_{i,i+1}$ and $k_{i-1,i}$, respectively. An efficient TPKA protocol for broadcasting is to employ authenticated Diffie-Hellman key agreement.

Each user U_i

(1) Generates a secret value r_i, and computes

$$x_i = \alpha^{f(ID_i)\cdot r_i} \pmod{n} \quad \text{and} \quad y_i = s_i \cdot h_i^{r_i \cdot f(x_i, T_i)} \pmod{n},$$

where f is a secure one-way function, T_i is a time-stamp indicating the current time and x_i acts as a verifiable ephemeral public key of U_i.

(2) Broadcasts (ID_i, x_i, y_i, T_i).

(3) Verifies for each $U_j, j = 1, \dots, m$, that

$$y_j^e \stackrel{?}{\equiv} ID_j \cdot x_j^{f(x_j, T_j)} \pmod{n}$$

and $|T' - T_j| < \Delta T$ where T' is the current time and ΔT is a margin due to time delay and clock inconsistencies.

(4) Establishes with U_{i-1} and U_{i+1} two secret values, respectively

$$k_{i-1,i} = x_{i-1}^{r_i} \pmod{n} \quad \text{and} \quad k_{i,i+1} = x_{i+1}^{r_i} \pmod{n}.$$

Round 2. Each user U_i

(1) Computes

$$\begin{aligned} v_i &= f(ID_i) \cdot k_{i,i+1} - f(ID_{i-1}) \cdot k_{i-1,i}, \\ X_i &= \alpha^{f(ID_i) \cdot k_{i,i+1}} \pmod n, \\ Y_i &= s_i \cdot h_i^{k_{i,i+1} \cdot f(X_i, v_i, c)} \pmod n, \end{aligned}$$

where $c = f(x_1, x_2, \ldots, x_m)$, linking the contributions of each participant of the first round to the second round.

(2) Broadcasts (ID_i, X_i, Y_i, v_i).

(3) Verifies whether

$$Y_j^e \stackrel{?}{\equiv} ID_j \cdot X_j^{f(X_j, v_j, c)} \pmod n$$

holds for each $U_j, j = 1, \ldots, m$.

(4) Secretly computes the common secret group key $\mathcal{Z}$ by

$$\begin{aligned} \mathcal{Z}_i &= v_{i-1} + 2 \cdot v_{i-2} + \ldots + (m-1) \cdot v_{i+1} + m \cdot f(ID_i) \cdot k_{i,i+1} \\ &= m \cdot f(ID_i) \cdot k_{i,i+1} + \sum_{j=1}^{m-1} j \cdot v_{i-j}, \end{aligned}$$

where $v_j = v_{m+j}$. The shared secret group key is $\mathcal{Z} = \sum_{j=1}^{m} f(ID_j) \cdot k_{j,j+1}$ and correspondingly if all messages and computations are correct, then for all users, $\forall\, i, j \in \{1, \ldots, m\}$, $\mathcal{Z} = \mathcal{Z}_i = \mathcal{Z}_j$.

(5) Computes the public group key $\mathcal{P}$ according to

$$\mathcal{P}_i = \prod_{j=1}^{m} X_j = \alpha^{\mathcal{Z}_i} \pmod n,$$

so that $\mathcal{P} = \mathcal{P}_i = \mathcal{P}_j$, $\forall\, i, j \in \{1, \ldots, m\}$ if the protocol succeeds.

By verifying that $\prod_{j=1}^{m} X_j \stackrel{?}{\equiv} \alpha^{\mathcal{Z}_i} \pmod n$, $\mathcal{P}$ and $\mathcal{Z}$ is implicitly authenticated due to that each X_i, $i = 1, \ldots, m$ is a factor of the public group key.

3.3. Confirmation and certification

In the last round, the participants collaboratively compute a signature for the public group key $\mathcal{P}$. The signature provides public verification of $\mathcal{P}$, and thus accordingly confirms the secret group key for the participants.

Round 3. Each user U_i

(1) Computes and broadcasts an individual signature of public group key

$$\mathcal{S}_i = s_i \cdot h_i^{\mathcal{Z}_i \cdot f(\mathcal{P}, \bar{T})} \pmod{n},$$

$\overline{T}$ is the average of T_j of each $U_j, j = 1, \ldots, m$ in the first round.

(2) Verifies the individual signatures for each U_j, $j = 1, \ldots, m$:

$$\mathcal{S}_j^e \stackrel{?}{\equiv} ID_j \cdot \mathcal{P}^{f(\mathcal{P}, \overline{T}) \cdot f(ID_j)} \pmod{n}.$$

(3) Computes the multi-party signature $\mathcal{S} = \prod_{j=1}^m \mathcal{S}_j \pmod{n}$ that corresponds to

$$\mathcal{S} = \left(\prod_{j=1}^{m} ID_j^d \right) \cdot \alpha^{d \cdot \mathcal{Z} \cdot f(\mathcal{P}, \overline{T}) \cdot \sum_{j=1}^m f(ID_j)} \pmod{n}.$$

Finally, the public group key and its certifying parameters is published: ($\mathcal{P}$, $\mathcal{S}$, $\overline{T}$, id) where $id = \{ID_i \| \, 1 \le i \le m\}$. $\mathcal{P}$ is certified according to

$$\mathcal{S}^e \stackrel{?}{\equiv} \left(\prod_{i=1}^{m} ID_i \right) \cdot \mathcal{P}^{\cdot f(\mathcal{P}, \overline{T}) \cdot \sum_{i=1}^m f(ID_i)} \pmod{n}.$$

Since the group key signature will detect errors and inequalities that would be detected in step 2, step 2 can be omitted.

3.4. Employment

By employing the ElGamal cryptosystem, outsiders are now able to encrypt messages by means of $\mathcal{P}$ so that only the legitimate group members associated to $\mathcal{P}$ are able to individually decrypt. A plaintext message M is encrypted by $a = \alpha^r \bmod n$ and $b = M \cdot \mathcal{P}^r \bmod n$, where r is a random number secretly chosen by the sender, and subsequently decrypted by $M = b \cdot a^{-\mathcal{Z}} \bmod n$. The group participants can correspondingly communicate secretly by $\mathcal{P}$, or by symmetric encryption by $\mathcal{Z}$.

4. Security analysis

In this section, we consider the security of the proposed cryptosystem according to the previously stated security requirements.

Theorem 4.1. *The secret key agreement protocol provides key secrecy, and is secure against passive attacks.*

P r o o f. The secret values $k_{i,i+1}$ are established due to authenticated Diffie-Hellman-based key exchange, where secrecy is achieved due to the Computational Diffie-Hellman Problem. That is, knowledge of $\alpha^x \bmod p$ and $\alpha^y \bmod p$ does not provide $\alpha^{xy} \bmod p$. After the first round, $k_{i,i+1}$ is known only by U_i and U_{i+1}. In the second round, U_i broadcasts v_i. Since this is the subtraction of two secrets, and assuming that Computational Diffie-Hellman Problem is hard, it is correspondingly infeasible for an adversary to deduce any secret $k_{j,j+1}$, thus, ensuring the secrecy of $\mathcal{Z}$. For a more formal security proof, see [7]. □

Theorem 4.2. *The employed signature scheme is secure.*

P r o o f. The signature scheme is analogous to the authentication scheme of [8], where the difficulty is to find two integers, x and y, so that $y_i^e \stackrel{?}{\equiv} ID_i \cdot x_i^{x_i}$ $(\bmod\ n)$ holds. The signature in the second round is represented by

$$\begin{aligned} X_i &= \alpha^{f(ID_i)\cdot k_{i,i+1}} \pmod{n}, \\ Y_i &= s_i \cdot h_i^{k_{i,i+1}\cdot f(X_i,v_i,c)} \pmod{n}, \end{aligned}$$

where $k_{i,i+1}$ represents a value that may be known by the other participants. The value of d is unknown. Given a signature Y_i, where the corresponding long-term secret user keys (s_i, h_i) are unknown to others than U_i, according to the Discrete Logarithm Problem, the factor $\alpha^{d\cdot f(ID_i)\cdot k_{i,i+1}\cdot f(X_i,v_i,c)}$ is computationally infeasible to obtain from Y_i since the value of d is unknown. Since s_i constitutes the other unknown factor of Y_i, it is thus infeasible to derive the secret user keys (s_i, h_i) from Y_i.

Since the factorization of n is unknown, it is correspondingly not feasible to obtain $ID_j^{(e^{-1})}$ $(\bmod\ n)$.

Attempts to forge a valid signature Y_i by raising it to the power of a different context c', $Y_i^{c'}$ will not provide a valid signature since (and as long) $ID_i^{c'}$ does not constitute a valid identity.

Given a secret user key $h_i = \alpha^{d\cdot f(ID_i)}$ $(\bmod\ n)$, it is likewise computationally infeasible to derive α^d according to the Discrete Logarithm Problem. □

A variant of this authentication scheme is found in [9] where the verification is according to

$$y_j^e \stackrel{?}{\equiv} ID_j \cdot x_j^{f(T)} \pmod{n},$$

where T is a time-stamp. This scheme is not resistant to the Extended Euclidian Algorithm attack [11]. If e and a are relatively prime, we can find two integers u, v, so that $e \cdot u = 1 + f(T) \cdot v$. An adversary can thus pick a valid ID_j, and compute $x_j = ID_j^v \pmod{n}$ and $y_j = ID_j^u \pmod{n}$. The attack succeeds in the scheme in [9] because

$$y_j^e = ID_j \cdot x_j^{f(T)} = \left(ID_j^u\right)^e = ID_j \cdot \left(ID_j^v\right)^{f(T)} \pmod{n}.$$

In our scheme, this attack is thwarted, since $y_j^e \stackrel{?}{\equiv} ID_j \cdot x_j^{f(x_j, T_j)} \pmod{n}$ and x_j is used as self-verifying parameter by raising it to a power determined by itself as parameter to the hash-function.

Theorem 4.3. *The protocol provides forward secrecy.*

P r o o f. The values of the group keys depend on the Diffie-Hellman values $k_{i,i+1}$, $1 \leq i \leq m$, established by each of the participants for each session. The secret group key is the sum of $f(ID_i) \cdot k_{i,i+1}$, $1 \leq i \leq m$, where we consequently assume that any $k_{i,i+1}$ is unknown to others than the group. Given $X_i = \alpha^{f(ID_i) \cdot k_{i,i+1}} \pmod{n}$ and $Y_i = s_i \cdot h_i^{k_{i,i+1} \cdot f(X_i, v_i, c)} \pmod{n}$, due to the Discrete Logarithm Problem, it is computationally infeasible to obtain the values of $f(ID_i) \cdot k_{i,i+1}$ and $k_{i,i+1} \cdot f(X_i, v_i, c)$ if s_i and h_i are disclosed. Thus, it is infeasible to obtain $k_{i,i+1}$. Forward secrecy is therefore achieved. □

Theorem 4.4. *The protocol is secure against substitution attacks.*

P r o o f. Assuming that the employed signature scheme is secure, it is computationally infeasible for an adversary A' to successfully forge any signature Y_i' for other sessions. Suppose A' knows the values $k_{i-1,i}'$, $k_{i,i+1}'$, v_i', X_i' and Y_i' of a former session *c'*. Because $k_{i-1,i}'$ and $k_{i,i+1}'$ refer to a former session, A' cannot set $k_{i-1,i}$ and $k_{i,i+1}$ of a new session to equal the former $k_{i-1,i}'$ and $k_{i,i+1}'$ due to the contributory nature of DH key agreement.

Thus, A' can neither control that v_i' will have a valid value, nor the session value *c'* to be equal subsequent sessions c, since c is unique for each session, collaboratively determined by inputs of all participants. Since Y_i' is cryptographically locked to v_i' and *c'*, replaying former messages will consequently be detected due to signature verification. Thus, the adversary cannot successfully replay and substitute messages with former ones. □

Theorem 4.5. *The proposed protocol provides unique group keys for each session.*

P r o o f. Each user U_i contributes equally to the values of the group keys by the secret $k_{i,i+1}$. Assuming that at least one participant selects a new and unique random r_i, then $k_{i,i+1}$ and $k_{i-1,i}$ will be unique, thereby making the group key pair unique. Moreover, assuming that secure signature scheme withstands substitution attacks, no single user can enforce old group keys to be re-established. Thus, key freshness is achieved. □

5. Conclusion

We have proposed an efficient and practical public group key cryptosystem, where groups containing any number of users can flexibly establish public/private key pairs that represent the groups. No online trusted third party is required neither for group key establishment nor group key certification. By means of public group keys, outsiders can address the group confidentially, and secure communication is provided internally among the group members. The signature scheme provides identity-based certification of public group keys and key establishment messages towards the corresponding group participants.

REFERENCES

[1] DESMEDT, Y.: *Society and group oriented cryptography: A new concept.* In: Advances in Cryptology–CRYPTO '87 (C. Pomerance, ed.), Lect. Notes Comput. Sci., Vol. 293, Springer-Verlag, Berlin, 1988, pp. 120–127.

[2] DESMEDT, Y.—FRANKEL, Y.: *Threshold cryptosystems.* In: Advances in Cryptology–CRYPTO '89 (G. Brassard, ed.), Lect. Notes Comput. Sci., Vol. 435, Springer-Verlag, New York, 1990, pp. 307–315.

[3] PEDERSEN, T.: *A threshold cryptosystem without a trusted party (Extended Abstract).* In: Advances in Cryptology–EUROCRYPT '91, (D. W. Davies, ed.), Lect. Notes Comput. Sci., Vol. 547, Spring-Verlag, Berlin, 1991, pp. 522–526.

[4] SAEEDNIA, S.—GHODOSI, H.: *A self-certified group-oriented cryptosystem without a combiner.* In: ACISP '99 (J. Pieprzyk, ed.), Lect. Notes Comput. Sci., Vol. 1587, Springer-Verlag, Berlin, 1999, pp. 192–201.

[5] GHODOSI, H.—SAEEDNIA, S.: *A modification to the self-certified group-oriented cryptosystem without a combiner.* Electronics Letters **37**, no. 2 (2001), 86–87.

[6] BURMESTER, M.—DESMEDT, Y.: *A secure and efficient conference key distribution system.* In: Advances in Cryptology–EUROCRYPT '94 (A. De Santis, ed.), Lect. Notes Comput. Sci., Vol. 950, Springer-Verlag, Berlin, 1994, pp. 275–286.

[7] JUST, M.—VAUDENAY, S.: *Authenticated multi-party key agreement.* In: Advances in Cryptology–ASIACRYPT '96 (K. Kim, ed.), Lect. Notes Comput. Sci., Vol. 1163, Springer-Verlag, Berlin, 1996, pp. 36–49.

[8] KOYAMA, K.—OHTA, K.: *Security if improved identity-based conference key distribution systems.* In: Advances in Cryptology–EUROCRYPT '88 (Ch. G. Günter, ed.), Lect. Notes Comput. Sci., Vol. 330, Springer-Verlag, Berlin, 1988, pp. 11–19.

[9] YANG, W.—SHIEH, S.: *Password authentication schemes with smart cards.* Computer & Security **18**, no. 8 (1999), 727–733.

[10] OKAMOTO, E.—TANAKA, K.: *Key distribution system based on identification information.* IEEE Journal on Selected Areas in Communications **7**, no. 4 (1989), 481–485.

[11] CHEN, K.—ZHONG, S.: *Attacks on the (enhanced) Yang-Shieh authentication.* Computer & Security **22**, no. 8 (2003), 725–727.

[12] BOYD, C.—MATHURIA, A.: *Protocols for Authentication and Key Establishment.* ISBN 3-540-43107-1, Springer-Verlag, Berlin, 2003.

[13] GIRAULT, M.: *Self-certified public keys.* In: Advances in Cryptology–EUROCRYPT '91 (D. W. Davies, ed.), Lect. Notes Comput. Sci., Vol. 547, Springer-Verlag, Berlin, 1991, pp. 490–497.

Received October 27, 2005

University of Agder
Grooseveien 36
4876 Grimstad
NORWAY
E-mail: sigurd.eskeland@uia.no