DE GRUYTER
OPEN

TATRA
MOUNTAINS
Mathematical Publications

DOI: 10.2478/tmmp-2014-0024
Tatra Mt. Math. Publ. **60** (2014), 47–55

NON-INTERACTIVE SECURE MULTIPARTY KEY ESTABLISHMENT

Sigurd Eskeland

ABSTRACT. Cryptographic schemes that provide establishment of secret keys among a number of participants are generally known as conference key establishment schemes and key broadcasting schemes. In any case, such protocols provide secure establishment of group-oriented cryptographic keys, but with the costs of multiple transmissions of key establishment messages and in some cases multiple secret user keys. In this paper, we present a simple and straightforward efficient non-interactive group-oriented key establishment scheme that provides off-line computation of secret group keys, without computations and transmissions of key establishment messages.

1. Introduction

A multitude of multiparty security schemes have been proposed during the years. The purpose of such schemes is to provide secure communication over insecure networks by secure establishment of a secret temporary group key that is shared among a group of users. Encrypting subsequent communication with the group key, secure communication can be obtained. Common for such schemes is that they are cryptographic *protocols* that specify a set of algorithms for computation of key establishment messages sent and received by the pertaining collaborating group participants, and their order of transmission. An intrinsic property about protocols is that all pertaining participants must be online simultaneously when group keys are established and later updated. Since the users collaboratively compute, receive and transmit key establishment messages, protocols are interactive. Some protocols require an online trusted key center or a group controller to coordinate the message flow, and are thus centralized. Other protocols are distributed and use no online key center.

2010 Mathematics Subject Classification: 94A60.
Keywords: key establishment, non-interactive key establishment, conference schemes.

The well-known Diffie-Hellman (DH) scheme is a basis for a large number of cryptographic schemes, including conference key agreement protocols, authentication schemes, password authentication schemes, signature schemes, public key encryption algorithms, and more. The DH scheme has an inherent constraint because only two public/private key pairs can constitute a shared key, since the public keys must act as a base and private keys as an exponent. Multiparty schemes that are based on the DH scheme overcome this constraint by using additional rounds of transmissions and computations.

In this paper, we propose a secure multiparty key agreement scheme that is presented in Section 4. It uses the user key generation scheme that is presented in Section 3. An essential advantage of this scheme over other key agreement schemes is that it is non-interactive, meaning that there are no key establishment messages required. Due to that it is non-interactive (in contrast to interactive round-oriented protocols), it is well fit for *ad hoc* multiuser situations and scenarios where group compositions are changed extensively.

2. Related work

In general, key establishment protocols define algorithms for computing key establishment messages, and the transmission sequences and dependencies of those key establishment messages, i.e., there are a number of rounds. Somewhat related to the presented scheme, are conference key agreement (CKA) protocols. This is an important class of cryptographic multiparty schemes that enable participants of a group to compute a shared key as a result of interaction among the participants. A common trait for all CKA protocols is that key establishment messages are being generated and transmitted by all involved users, in contrast to broadcast encryption schemes discussed next. CKA may have or have not a group leader. Moreover, CKA schemes are *contributory*, meaning that each participant contributes to the value of the group key. Contributory key establishment schemes are usually based on the Diffie-Hellman two-party key agreement scheme [1]. Examples of some CKA schemes are found in [2]–[5]. B o y d [6, Ch. 6] provides a good survey.

As noted, protocols define the transmission sequences and dependencies of those key establishment messages. In contrast, the key agreement scheme presented in [7] has no transmissions of key establishment messages. In this scheme, group keys are computed as a function of a private key and public keys. It uses primes as public keys as exponents modulo a composite number. It works as follows: A trusted authority computes a composite number $n = p \cdot q$, where p and q are large secret primes. A *secret* base g of high index is chosen. Let $\mathcal{U}$ denote a group of an arbitrary number of users, and let $T \subseteq \mathcal{U}$ denote

an arbitrary user subset. Each user $P_i \in \mathcal{U}$ is assigned a private key $g_i = g^{p_i} \pmod n$, where p_i, p_j are public and relatively prime for all users $P_i, P_j \in \mathcal{U}$, $i \neq j$. For a set of users $T \subseteq \mathcal{U}$, a shared group key is computed as $g_T = g^{p_T} \pmod n$, where $p_T = \prod_{i \in I_T} p_i$ and $I_T = \{j \,|\, P_j \in T\}$. Thus, each user can compute a group key K_T for any user subset he or she is a member of as $g_i^{p_{i,T}} \pmod n$, where $p_{i,T} = \prod_{\substack{j \in I_T \\ i \neq j}} p_j$.

The scheme is vulnerable to the so-called Euclidean attack as follows: For any two p_i, p_j, using the Extended Euclidean Algorithm, the two numbers a and b can be computed, so that $p_i \cdot a + p_j \cdot b = 1$, since p_i, p_j are relatively prime. Thus, two users $P_i, P_j \in \mathcal{U}$ can establish the secret base $g \equiv g_i^a \cdot g_j^b \pmod n$, and hence, the group key for any user composition. The scheme is therefore not resistant to an attack involving two users, and is thus said to be 1-resilient.

3. A user key generation scheme for non-interactive secure multiparty computations

In this section, we present a new user key generation scheme, whose user keys are part of the non-interactive multiparty key agreement scheme presented in Section 4.

Initialization. A Trusted Authority (TA) is required for computing the long-term user keys. Note that the TA is not involved in group key computation (Section 4). Initially, the TA provides the following computations:

- The TA selects two large secret primes p and q. Let $n = p \cdot q$ be public.
- Let α be a public base that has a high order in $\mathbb{Z}_{\phi(n)}$, where $\phi(n) = (p-1)(q-1)$. For convenience, let $\phi' = \phi(n)$.
- The TA randomly selects a large secret number $p' \in \mathbb{Z}_{\phi'}$.

User key generation. Let $\mathcal{U} = \{P_1, P_2, \ldots\}$ denote a group of an arbitrary number of users that each will be assigned a public/private key pair. For each participant $P_i \in \mathcal{U}$, the TA carries out the following tasks:

- Assign a unique public identity id_i.
- Randomly generate a secret unique number $v_i \in \mathbb{Z}_{\phi'}$.
- Let f be a secure one-way function. Compute the secret $z_i = f(id_i) \bmod \phi'$.
- Compute the private key $x_i = z_i\, p' + v_i\, \phi'$.
- Discard the secret values v_i, z_i.

Concerning step 3, it is crucial that $||f(id_i)|| > ||n|| + b$, that is, the number of bits of f should exceed the size of n with at least $b = 200$ bits.

3.1. Security considerations

Identity-based public keys. Note that implicitly, a public key $pk_i = f(id_i)$ is computed as a result of the identity id_i and the one-way function f. The fact that it is convenient to use meaningful identities is the main motivation for incorporating identities in the presented scheme. This is because the authenticity of the public keys is guaranteed implicitly, and less storage space and transmission bandwidth is required, since the number of bits for storing id_i may be considerable less than storing pk_i.

The composite integer n should be at least in the order of 1024 bytes, and the output size of f should exceed this with at least 200 bits. This condition guarantees that z_i is protected and not disclosed given pk_i, since ϕ' is unknown.

Although there exist hash functions supporting variable output sizes, like the SHA-3 candidate Skein, variable output sizes can be achieved by using standard hash functions like AES by concatenation, as $H(1\,|M)\,|\,H(2\,|M)\,|\,H(3\,|M)\,|\ldots$

These considerations aside, public keys could alternatively be generated as large random strings R_i that correspondingly must exceed the size of n with at least 200 bits. The secret value z_i is correspondingly determined by $z_i = R_i \bmod \phi'$. Since we use user identities to implicitly establish pk_i, we therefore do not consider other security issues related to the aspect of identity-based encryption other than assuming that f is secure to avoid collisions.

Considerations of the key composition values. Due to the requirement $||pk_i|| > ||n|| + b$, public key values can be formulated as $pk_i = z_i + w_i\,\phi'$, where w_i would be unknown to all but the TA, and the product $w_i \cdot \phi'$ implicitly conceals the secret z_i correspondingly.

Private keys are established as $x_i = z_i p' + v_i \phi'$, where the key composition numbers $\theta' = \{p', \phi', (v_i, z_i, \,|\, P_i \in \mathcal{U})\}$ are unknown to all but the TA. The purpose of the secret product $v_i \cdot \phi'$ is to algebraically conceal the secret product $z_i \cdot p'$, and (z_i, p') individually.

The public and private user keys are to be used as exponents modulo n. Therefore note that the terms containing the factor ϕ' in $x_i = z_i\,p' + v_i\,\phi'$ and $pk_i = f(id) = z_i + w_i\,\phi'$ are eliminated.

3.2. Security analysis

The security is based on the secrecy of the key composition numbers $\theta = \{p', \phi', (v_i, w_i, z_i \,|\, P_i \in \mathcal{U})\}$ that constitute $(x_i, pk_i \,|\, P_i \in \mathcal{U})$. It is crucial that all numbers in θ are prevented from disclosure, cf. Theorem 1, since this is directly tied to the preservation of security requirements described in Section 4.2. In the following analysis we focus on the secrecy of the elements in θ.

Theorem 1. *It is prevented that any of the secret key composition numbers in* $\theta = \{p', \phi', (v_i, w_i, z_i, \,|\, P_i \in \mathcal{U})\}$ *can be revealed from any of the long-term user keys* $(x_i, pk_i \,|\, P_i \in \mathcal{U})$.

P r o o f. Public/private user key pairs constitute equation systems

$$\begin{aligned} x_i &= z_i\, p' + v_i\, \phi' \\ pk_i &= \ \ z_i \ \ + w_i\, \phi' \end{aligned}$$

where the generic composition numbers (ϕ', p') are unknown, and (z_i, v_i, w_i) are unknown and unique for $P_i \in \mathcal{U}$. In general, k user key pairs result in an equation system of $k' = 2k$ equations. Hence, for each user key pair added into the equation system, there are 3 more unknowns added to the system (i.e., z_i, v_i, w_i, disregarding the product $z_i \cdot p'$). Any such equation system is hence underdefined and results in infinitely many solutions. The key composition numbers are therefore algebraically prevented from disclosure given any set of user keys.

The secrecy of ϕ' is moreover related to the difficulty of solving the factorization of n. A machine M_1 that could effectively factorize n would enable an attacker to compute $\phi' = (p-1) \cdot (q-1)$, whose disclosure would break the security. However, the Factorization Problem is known to be computationally infeasible, preventing disclosure of ϕ'.

It is known that the Discrete Logarithm Problem (DLP) modulo a large composite number n is as difficult as the DLP modulo a large prime, since it suffices to factorize n and then solve DLP each prime factor [8]. Using public keys as exponents to a base β modulo n, due to the DLP, it is computationally infeasible to disclose $z_i \in \theta$ given $\beta^{pk_i} \equiv \beta^{z_i} \pmod{n}$. Correspondingly, using private keys as exponents to any base β modulo n, due to the DLP, it is computationally infeasible to disclose $p' \cdot z_i$ given $\beta^{x_i} \equiv \beta^{p' \cdot z_i} \pmod{n}$. In general, it is computationally infeasible to deduce elements in θ from numbers containing user keys as exponents because of the difficulty of solving the DLP.

Therefore, the secret key composition numbers in θ are prevented from being disclosed according to Theorem 1. □

A note on the Euclidean attack. Due to the secrecy of ϕ', the Euclidean attack described in Section 2 cannot be applied on sets of public/private user keys. In contrast, the attack could be carried out for two known values $(w_i, w_j \,|\, P_i, P_j \in \mathcal{U})$, which are concealed as shown. Attempting to use the Extended Euclidean Algorithm on two private keys (x_i, x_j) that are relatively prime results in (a, b), where $x_i \cdot a + x_j \cdot b = 1$. Using these values as exponents to a base β modulo n as an attempt to eliminate them results in

$$\beta^{x_i \cdot a + y_j \cdot b} \equiv \beta^{(z_i\, p' + v_i\, \phi') \cdot a + (z_j\, p' + v_j\, \phi') \cdot b} \equiv \beta^{z_i \cdot p' \cdot a + z_j \cdot p' \cdot b} \equiv \beta^{p' \cdot u} \pmod{n},$$

where $u \neq 1$ and unknown due to the secret key composition numbers (z_i, z_j). The same is the case for public keys. Hence, the Euclidean attack does not work on the presented scheme.

4. Multiparty key agreement without user interaction

In this section, we propose a non-interactive key agreement scheme based on the user key generation scheme presented in Section 3. Several variants of this basic agreement scheme are possible. The proposed variant in this section is symmetric in the sense that there is no online key center that initiates a conference and that provides ephemeral key establishment data for the conference users. For an arbitrary user group $T \subseteq \mathcal{U}$, the shared group key K_T is computed solely as a function of that group's user composition using the private key of a given user $P_i \in T$, and the identities of $P_j \in T \backslash \{P_i\}$ as input values. Therefore, no online key center is required to compute K_T.

Each $P_i \in T$ computes the group-specific key as

$$\begin{aligned} K_T = K_{T,i} &= \alpha^{x_i \cdot \prod_{j \mid P_j \in (T - \{P_i\})} f(id_j)} \pmod{n} \\ &= \alpha^{p' \cdot \prod_{j \mid P_j \in T} z_j} \pmod{n}. \end{aligned}$$

There is no communication required to compute group keys K_T, which are computed as a function of the long-term user keys of the user coalition $T \subseteq \mathcal{U}$. Data to be communicated confidentially is encrypted by means of a secure symmetric key cryptographic algorithm using K_T as the secret cryptokey.

4.1. Correctness

Since the long-term user keys are used as exponents modulo n, terms containing the secret factor ϕ' are eliminated, since computing powers modulo n are equivalent to exponentiations in the cyclic group $\mathbb{Z}_{\phi'}$:

$$\begin{aligned} K_T \equiv K_{T,i} &\equiv \alpha^{x_i \cdot \prod_{j \mid P_j \in (T - \{P_i\})} f(id_j)} \pmod{n} \\ &\equiv \alpha^{x_i \cdot \prod_{j \mid P_j \in (T - \{P_i\})} pk_j} \pmod{n} \\ &\equiv \alpha^{(z_i \cdot p' + v_i \cdot \phi') \cdot \prod_{j \mid P_j \in (T - \{P_i\})} (z_i + w_i \cdot \phi') \pmod{\phi'}} \pmod{n} \\ &\equiv \alpha^{p' \cdot z_i \cdot \prod_{j \mid P_j \in (T - \{P_i\})} z_j} \pmod{n} \\ &\equiv \alpha^{p' \cdot \prod_{j \mid P_j \in T} z_j} \pmod{n} \\ &\equiv \alpha^{p' \cdot z_T} = h^{z_T} \pmod{n}, \end{aligned}$$

where $z_T = \prod_{j \mid P_j \in T} z_j$ and $h = \alpha^{p'}$. Hence, the correctness of the scheme is provided as shown.

4.2. Security requirements

We assume that there exists a set $\mathcal{U}$ of an arbitrary number of users. A group key K_T is computed as a function of the long-term user keys of any user subset $T \subseteq \mathcal{U}$. We make the assumption of an adversary that is equivalent with

a user coalition $A \subset \mathcal{U}$, where $T \subseteq \mathcal{U}$ and $A \cap T = \emptyset$. Hence, for any $P_i \in A$, then $P_i \notin T$. Consequently, this is a stronger assumption than an adversary A', where $A' \cap \mathcal{U} = \emptyset$.

Adversary capabilities. We assume that A may hold the following information:

1. The private user keys x_i for each $P_i \in A$.
2. The group keys K_{T^*}, where $T^* \subset \mathcal{U}$ and $A \cap T^* \neq \emptyset$, since K_{T^*} is easy to compute using $(x_i \mid P_i \in A)$. In contrast, $A \cap T = \emptyset$ as noted above.
3. The public user key pk_i for each $P_i \in \mathcal{U}$.

The security of the scheme is based on the following security requirements:

Security Requirement 1. *Secrecy of private keys.* It must be computationally infeasible for an adversary $A \subset \mathcal{U}$ to reestablish the private user key x_i of $P_i \notin A$.

Security Requirement 2. *Secrecy of group keys.* It must be computationally infeasible for an adversary $A \subset \mathcal{U}$, where $T \subseteq \mathcal{U}$ and $A \cap T = \emptyset$, to compute the group key K_T.

As pointed out, there is no communication required to compute group keys K_T which are computed as a function of the long-term user keys of the user coalition $T \subseteq \mathcal{U}$. Data to be communicated confidentially is encrypted by means of a secure symmetric key cryptographic algorithm using the secret group key as cryptokey. Assuming that the symmetric key cryptographic algorithm used is secure, we do not consider attacks on the communicated encrypted data.

4.3. Security considerations

Concerning the stated security requirements, it is crucial that the secrecy of the key composition numbers of θ is preserved. Moreover, the secrecy of the power $h = \alpha^{p'}$ is crucial with regard to Security Requirement 2. An adversary that is able to deduce h can easily compute $K_T = h^{\prod_{j \mid P_j \in T} pk_j} \pmod{n}$.

4.4. Security analysis

In this section, we show that the security of the scheme is in agreement with the stated security requirements.

Definition 1. The Discrete Logarithm Problem (DLP) is closely related to the Computational Diffie-Hellman (CDH) problem. The CDH problem is as follows: Let p be a large prime and α a primitive root to p. Then given two randomly chosen values $\alpha^x \pmod{p}$ and $\alpha^y \pmod{p}$, find $\alpha^{xy} \pmod{p}$. A variation of CDH is the Static DH Problem, where α^x and α^{xy} are known. The difficulty is to find α^y, which is equivalent to the hardness of the CDH problem [9].

Theorem 2. *It is computationally infeasible to reveal* $h = \alpha^{p'}$.

P r o o f. The secrecy of the exponent p' is preserved according to Theorem 1. Using (x_i, pk_i) as exponents to α gives $\alpha^{x_i} = \alpha^{p' \cdot z_i}$ and $\alpha^{pk_i} = \alpha^{z_i}$, where p' and z_i are unknown. Finding $h = \alpha^{p'}$ given $\alpha^{p' \cdot z_i}$ and α^{z_i} is equivalent of solving the Static DH Problem. Since this problem is assumed to be hard to solve, the secrecy of h is preserved as shown. □

Theorem 3. *Security Requirement 1 is preserved.*

P r o o f. *Secrecy of private keys.* The user keys are computed independently of each other, since $(v_i, w_i \,|\, P_i \notin \mathcal{U})$ are randomly generated, and the value of $(z_i \,|\, P_i \notin \mathcal{U})$ is randomized due to f and the unknown ϕ'. Thus, in order for A to deduce private keys $(x_i \,|\, P_i \notin A)$, knowledge of key composition numbers in θ is required. According to Theorem 1 (Section 3.2), the secrecy of the key composition numbers in θ is preserved. Therefore, Security Requirement 1 is preserved. □

Theorem 4. *Security requirement 2 is preserved.*

P r o o f. *Secrecy of group keys.* The correctness outline of K_T (Section 4.1) shows how it is constituted of elements in θ, and implicitly constituted of the unknown value $h = \alpha^{p'}$. According to Theorem 1 (Section 3.2), the secrecy of the key composition numbers in θ is preserved. According to Theorem 2, the secrecy of $h = \alpha^{p'}$ is preserved. Thus, computation of K_T using h or elements in θ is prevented.

According to Security requirement 2, it must be computationally infeasible for $A \subset \mathcal{U}$ (where $T \subseteq \mathcal{U}$, $A \cap T = \emptyset$) to compute the group key K_T. In agreement with the adversary assumptions in Section 4.2, let $T^* \subset \mathcal{U}$, $A \cap T^* \neq \emptyset$. There exists group keys K_{T^*} within $T^* \supset A$ that are easy for A to compute:

$$K_{T^*} = \alpha^{x_i \cdot \prod_{j \,|\, P_j \in (T^* - \{P_i \in A\})} pk_j} = \alpha^{p' \cdot \prod_{j \,|\, P_j \in T^*} z_j} = \alpha^{p' \cdot z_{T^*}}$$

Let $T^{**} \subset \mathcal{U}$ so that $T^* \cap T^{**} = \emptyset$ and $T = T^* \cup T^{**}$. It is easy for $P_i \in A$ to compute

$$L_{T^{**}} = \alpha^{\prod_{j \,|\, P_j \in T^{**}} pk_j} = \alpha^{\prod_{j \,|\, P_j \in T^{**}} z_j} = \alpha^{z_{T^{**}}}$$

Finding $K_T = \alpha^{p' \cdot z_T}$ given $K_{T^*} = \alpha^{p' \cdot z_{T^*}}$ and $L_{T^{**}} = \alpha^{z_{T^{**}}}$ is equivalent of solving the Static DH Problem. Since this problem is assumed to be hard to solve, the secrecy of the exponent K_T is preserved as shown. Therefore, Security Requirement 2 is preserved. □

5. Conclusion

Many conference key establishment protocols that are proposed in the literature provide secure establishment of group-oriented cryptographic keys. Such schemes have the cost of multiple computations and transmissions of key establishment messages. In this paper, we have presented a novel public/private user key generation scheme that allows public keys to be used as exponents. The user keys are computed in such a way that it is computationally infeasible to deduce the secret internal key values as shown in the security analysis. We have moreover presented a non-interactive group-oriented key establishment scheme that uses the mentioned user key generation scheme, which provides off-line computation of secret group keys. Since there are no interactive rounds of computation and transmission of key establishment data, as it is the case of ordinary cryptographic protocols, it provides efficient multiparty computations of secret group keys. Future work could be to modify the scheme into a one-to-many public key cryptographic algorithm.

REFERENCES

[1] DIFFIE, W.—HELLMAN, M.: *New directions in cryptography*, IEEE Trans. Inform. Theory **22** (1976), 644–654.

[2] INGEMARSSON, I.—TANG, D.—WONG, C.: *A conference key distribution system*, IEEE Trans. Inform. Theory **28** (1982), 714–720.

[3] BURMESTER, M.—DESMEDT, Y.: *A secure and efficient conference key distribution system*, in: Advances in Cryptology—EUROCRYPT '94 (A. De Santis, ed.), Workshop on the Theory and Appl. of Cryptographic Techniques, Perugia, Italy, 1994, Lecture Notes in Comput. Sci., Vol. 950, Springer, Berlin, 1994, pp. 275–286.

[4] STEINER, M.—TSUDIK, G.—WAIDNER, M.: *Diffie-Hellman key distribution extended to group communication*, in: Proc. of the 3rd ACM Conf. on Comput. and Commun. Security—CCS '96, ACM, New York, 1996, pp. 31–37.

[5] ATENIESE, G.—STEINER, M.—TSUDIK, G.: *Autenticated group key agreement and friends*, in: Proc. of the 4th ACM Conf. on Comput. and Commun. Security—CCS '98, ACM, New York, 1998, pp. 17–26.

[6] BOYD, C.—MATHURIA, A.: *Protocols for Authentication and Key Establishment*, in: Information Security and Cryptography, Springer, Berlin, 2003.

[7] FIAT, A.—NAOR, M.: *Broadcast encryption*, in: Advances in Cryptology—CRYPTO '93, Lecture Notes in Comput. Sci., Vol. 773, Springer, Berlin, 1994, pp. 480–491.

[8] BACH, E.: *Discrete Logarithms and Factoring*, University of California at Berkeley, Berkeley, CA, USA, 1984.

[9] BROWN, D. R. L.—GALLANT, R. P.: *The Static Diffie-Hellman Problem*, Cryptology ePrint Archive, Report 2004/306, 2004.

Received September 22, 2014

Kirkeveien 5 A
N-4631 Kristiansand
NORWAY
E-mail: sigurd@inbox.com