Information Processing Letters 109 (2009) 842–845

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

A one-round secure message broadcasting protocol through a key sharing tree

Takaaki Mizuki [a,*], Takuya Sato [b], Hideaki Sone [a]

[a] Cyberscience Center, Tohoku University, Aramaki-Aza-Aoba 6-3, Aoba-ku, Sendai 980-8578, Japan
[b] Sone Lab., Graduate School of Information Sciences, Tohoku University, Aramaki-Aza-Aoba 6-3, Aoba-ku, Sendai 980-8578, Japan

ARTICLE INFO

Article history:
Received 19 May 2005
Available online 5 April 2009
Communicated by Y. Desmedt

Keywords:
Cryptography
Design of algorithms
Key sharing graph
Multiparty secret key exchange
Information-theoretic security

ABSTRACT

A key sharing graph is one in which each vertex corresponds to a player, and each edge corresponds to a secret key shared by the two players incident with the edge. Assume that, given a key sharing graph which contains a spanning tree, any designated player wishes to broadcast a message to all the other players securely against an eavesdropper. This can be easily done by flooding the message on the tree using the one-time pad scheme. However, the number of communication rounds in such a protocol is equal to the height of the tree. This paper provides another efficient protocol, which has exactly one communication round, i.e., we give a non-interactive protocol.

1. Introduction

Assume that there are n players $P_1, P_2, \dots, P_n$ where $n \geqslant 2$, and that there is an eavesdropper, Eve. Consider a situation in which several pairs of players have shared one-bit secret keys. We regard each player P_i as a vertex i in a graph G, and regard each one-bit secret key $k_{ij} \in \{0,1\}$ shared by players P_i and P_j as an edge ij in the graph G. (Refer to [5] for the graph-theoretic terminology.) Such a graph G is called a *key sharing graph* (e.g. [3,8]).

We assume that, given a key sharing graph G, all players and Eve know the shape of G, while (the value of) every secret key is private only to the two players who share it. Furthermore, assume that there is only a public authenticated channel (and hence there is no private channel); therefore, all communication among players is by public broadcast and is overheard by Eve.

The problem considered in this paper is quite simple and commonplace: we want to design a protocol which, given a key sharing graph G, can make any designated player broadcast a one-bit message $m \in \{0,1\}$ to all the other players securely against Eve. In other words, we want to make use of a key sharing graph G in order for all players to securely exchange a one-bit message m. Such a obtained message can be used for various purposes such as a conference key [2]. Also, the problem above often arises and becomes important, when one wants to extend some 2-player secret key exchange protocol to an n-player secret key exchange protocol for $n \geqslant 3$, namely a multiparty protocol (e.g. [6–8]).

If a given key sharing graph is not connected, then one can easily notice that secure message broadcasting is impossible; thus, we assume hereafter that any key sharing graph G appearing in this paper is connected, i.e., there exists a path between every pair of vertices in G.

1.1. The flooding protocol

The problem mentioned above is easy to solve by the following *flooding protocol*, which has been extensively used as a subprotocol when multiparty secret key exchange protocols were built (e.g. [6–8,13,14,18]).

Without loss of generality, we assume that the designated player is P_1, i.e., player P_1 wishes to broadcast a message m to all the other players $P_2, P_3, \dots, P_n$ (throughout the paper). Given a key sharing graph G, player P_1 can

* Corresponding author.
E-mail address: tm-paper@rd.isc.tohoku.ac.jp (T. Mizuki).

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2009.04.004

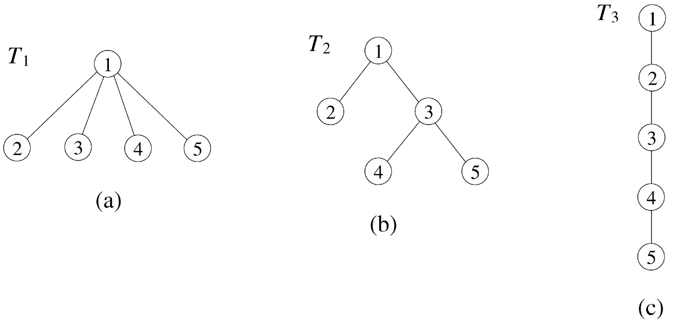

Fig. 1. Examples of key sharing trees.

securely send a one-bit message m to all the other players by flooding on the graph G (using the one-time pad scheme [15]), as follows.

1. Player P_1 sends the message m securely to each neighbor P_i of P_1 using the secret key k_{1i} shared by P_1 and P_i as a one-time pad. More precisely, P_1 announces $m \oplus k_{1i}$ publicly for each neighbor P_i of P_1, and then each neighbor P_i obtains the message m by computing $(m \oplus k_{1i}) \oplus k_{1i}$.
2. Every player P_i who has now obtained the message m sends it to each neighbor of P_i not yet obtaining m in the same way as above.
3. Repeat step 2 until all the players obtain the message m (remember that the key sharing graph G was assumed to be connected).

Thus, the message m is spread over along a spanning tree T rooted at P_1, which is an induced subgraph of G. Therefore, the number of communication rounds in the flooding protocol is equal to the height of the tree T.

Since a spanning tree suffices for secure message broadcasting as above, we hereafter consider only a key sharing graph which is just a tree, i.e., we address only such a *key sharing tree* for simplicity in exposition. (Thus, a key sharing tree is a "minimal" model in a sense because any non-connected key sharing graph never establishes secure message broadcasting as mentioned before.)

1.2. Our result

As seen above, the flooding protocol is simple and useful; furthermore, its round complexity equals the height of a given key sharing tree. Now, take three key sharing trees T_1, T_2 and T_3 depicted in Fig. 1 as examples. When the flooding protocol is executed, one can easily see that the numbers of communication rounds required for T_1, T_2 and T_3 are equal to one, two and four, respectively. Thus, one might feel that the key sharing tree T_1 is preferable to T_2 and T_3. However, we will show that it is not the case, concerning round complexity, as mentioned below.

In this paper, we will provide another new protocol, which is also simple and quite efficient. Specifically, our protocol has exactly one communication round, i.e., it is a non-interactive protocol. Therefore, for instance, whichever key sharing tree T_1, T_2 or T_3 is given, our protocol terminates after only one communication round. In this sense, these three key sharing trees T_1, T_2 and T_3 are equivalent; round complexity does not depend on the shapes of key sharing trees.

Our one-round secure message broadcasting protocol will be shown in Section 2.

1.3. Related works

The most famous appearance of key sharing graphs is in the dining cryptographers (or DC-nets) problem [3, 11]; given a key sharing graph, all players wish to accomplish anonymous message transmission, i.e., they wish to securely compute the parity of all their secret bits in a non-committed format. Somewhat related to key sharing graphs is the concept of the "key graphs" which are used for group key management systems [17].

As another direction of applying graph theory to cryptography, we point out that a graph often offers an useful model of communication channels. For example, graph theory plays an important role when one designs cryptographic protocols over partial broadcast channels [9], multicast channels [10,16], neighbor network channels [4] and so on. Furthermore, trade-offs between topology of communication channels and performance of secure computations have been much investigated [1,12].

2. Our one-round protocol

In this section, we provide our secure message broadcasting protocol, which terminates within one communication round. In Section 2.1, we give some examples of execution of our one-round protocol in order to exhibit the idea behind it. In Section 2.2, we present the description of our protocol. In Section 2.3, we verify the secrecy of our protocol.

2.1. Examples

Consider the key sharing tree T_3 depicted in Fig. 1(c) again. Remember that the number of communication rounds required for T_3 in the flooding protocol is exactly four (which equals the height of T_3). Now, instead of using the flooding protocol, let players P_1, P_2, P_3, P_4 do the followings simultaneously:

- P_1 announces $c_2 = m \oplus k_{12}$;
- P_2 announces $c_3 = k_{12} \oplus k_{23}$;

- P_3 announces $c_4 = k_{23} \oplus k_{34}$;
- P_4 announces $c_5 = k_{34} \oplus k_{45}$.

Then, players P_2, P_3, P_4, P_5 can obtain the message m as follows:

- P_2 computes $c_2 \oplus k_{12}$;
- P_3 computes $c_2 \oplus c_3 \oplus k_{23}$;
- P_4 computes $c_2 \oplus c_3 \oplus c_4 \oplus k_{34}$;
- P_5 computes $c_2 \oplus c_3 \oplus c_4 \oplus c_5 \oplus k_{45}$.

Thus, one-round communication achieves secure message broadcasting for the key sharing tree T_3 (its secrecy will be verified in Section 2.3).

Before going to the second example, we define an "internal" player. We say that, for a key sharing tree T, a player P_i, $2 \leqslant i \leqslant n$, is *internal* if P_i has two or more secret keys, i.e., two or more edges are connected to the vertex i. Note that players P_2, P_3 and P_4 (who made announcements) in the example above (Fig. 1(c)) are internal. Furthermore, it should be noted that, while P_1 announced the exclusive-or of the message m and the secret key k_{12} shared with her child P_2, every internal player announced the exclusive-or of the secret key shared with her parent and one shared with her child.

For the second example, consider the key sharing tree T_2 depicted in Fig. 1(b). Note that P_3 is the only internal player. Furthermore, notice that each of P_1 and P_3 has exactly two children. Let players P_1 and P_3 do the followings simultaneously:

- P_1 announces $c_2 = m \oplus k_{12}$ and $c_3 = m \oplus k_{13}$;
- P_3 announces $c_4 = k_{13} \oplus k_{34}$ and $c_5 = k_{13} \oplus k_{35}$.

Then, players P_2, P_3, P_4, P_5 can obtain the message m as follows:

- P_2 computes $c_2 \oplus k_{12}$;
- P_3 computes $c_3 \oplus k_{13}$;
- P_4 computes $c_3 \oplus c_4 \oplus k_{34}$;
- P_5 computes $c_3 \oplus c_5 \oplus k_{35}$.

Thus, one-round communication achieves secure message broadcasting also for the key sharing tree T_2. It should be noted that every player P_i other than P_1 obtained the message m by adding the secret key shared with her parent to the sum of all values announced by the players on the path between P_1 and the parent of P_i (modulo 2).

By carrying the idea further, one can easily build a one-round secure message broadcasting protocol as in the succeeding subsection.

2.2. Description of our protocol

Given a key sharing tree T, in order for player P_1 to securely broadcast a message m, our protocol proceeds as follows.

1. Execute the following (a) and (b) simultaneously.
   (a) Player P_1 announces $c_i = m \oplus k_{1i}$ for each child P_i of P_1.
   (b) Every internal player P_i, whose parent is P_ℓ, announces $c_j = k_{\ell i} \oplus k_{ij}$ for each child P_j of P_i.
2. Every player P_i other than P_1 obtains the message m by computing $c_{v_1} \oplus c_{v_2} \oplus \cdots \oplus c_{v_\ell} \oplus c_i \oplus k_{v_\ell i}$, where $1 \to v_1 \to v_2 \to \cdots \to v_\ell$ is the path between P_1 and the parent P_{v_ℓ} of P_i.

One can observe that all the players obtain the message m after the protocol above terminates. Thus, our protocol achieves message broadcasting for any key sharing tree within one communication round. Its secrecy will be verified in the succeeding subsection.

2.3. Secrecy of our protocol

In the sequel, we use the following notation: $p(i)$ denotes the index of the parent of P_i (in a key sharing tree), and $p^\ell(i)$ with $\ell \geqslant 2$ means $p(p^{\ell-1}(i))$ recursively where $p^1(i) = p(i)$; $\bar{x} \overset{\text{def}}{=} 1 - x$ for a bit $x \in \{0,1\}$; and $\overline{X} \overset{\text{def}}{=} (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_\ell)$ for an ℓ-bit sequence $X = (x_1, x_2, \ldots, x_\ell) \in \{0,1\}^\ell$.

Fix a key sharing tree T. Then, both a message $m \in \{0,1\}$ and a *key-value*

$$(k_{p(2)2}, k_{p(3)3}, \ldots, k_{p(n)n}) \in \{0,1\}^{n-1}$$

together determine the "conversation" $(c_2, c_3, \ldots, c_n)$ announced publicly in step 1 of the protocol. This can be expressed by a mapping $\mathsf{conv} : \{0,1\} \times \{0,1\}^{n-1} \to \{0,1\}^{n-1}$ such that

$$\mathsf{conv}\big(m, (k_{p(2)2}, k_{p(3)3}, \ldots, k_{p(n)n})\big) = (c_2, c_3, \ldots, c_n),$$

where

$$c_i = \begin{cases} m \oplus k_{1i} & \text{if } p(i) = 1; \\ k_{p^2(i)p(i)} \oplus k_{p(i)i} & \text{otherwise} \end{cases}$$

for every i, $2 \leqslant i \leqslant n$. We say that an $(n-1)$-bit sequence $(c_2, c_3, \ldots, c_n) \in \{0,1\}^{n-1}$ is a *conversation* if there exist $m \in \{0,1\}$ and $K \in \{0,1\}^{n-1}$ such that $\mathsf{conv}(m, K) = (c_2, c_3, \ldots, c_n)$. Concerning the mapping conv defined above, we have the following two lemmas.

Lemma 1. *For any $m \in \{0,1\}$ and $K, K' \in \{0,1\}^{n-1}$ with $K \neq K'$, $\mathsf{conv}(m, K) \neq \mathsf{conv}(m, K')$.*

Proof. Let

$$K = (k_{p(2)2}, k_{p(3)3}, \ldots, k_{p(n)n})$$

and

$$K' = (k'_{p(2)2}, k'_{p(3)3}, \ldots, k'_{p(n)n})$$

satisfy $k_{p(i)i} \neq k'_{p(i)i}$ for some i, $2 \leqslant i \leqslant n$. Let $\mathsf{conv}(m, K) = (c_2, c_3, \ldots, c_n)$, and let $\mathsf{conv}(m, K') = (c'_2, c'_3, \ldots, c'_n)$. Let ℓ be such that $p^\ell(i) = 1$.

When $\ell = 1$, i.e., $p(i) = 1$, we have $k_{1i} \neq k'_{1i}$ and hence

$$c_i = m \oplus k_{1i} \neq m \oplus k'_{1i} = c'_i.$$

Therefore, $\mathsf{conv}(m, K) \neq \mathsf{conv}(m, K')$, as desired.

Assume that $\ell \geqslant 2$. If $k_{p^2(i)p(i)} = k'_{p^2(i)p(i)}$, then by $k_{p(i)i} \neq k'_{p(i)i}$ we have

$$c_i = k_{p^2(i)p(i)} \oplus k_{p(i)i} \neq k'_{p^2(i)p(i)} \oplus k'_{p(i)i} = c'_i;$$

thus one may assume that $k_{p^2(i)p(i)} \neq k'_{p^2(i)p(i)}$. Similarly, one may assume that $k_{p^{j+1}(i)p^j(i)} \neq k'_{p^{j+1}(i)p^j(i)}$ for every j, $2 \leqslant j \leqslant \ell - 1$, and hence we have $k_{1p^{\ell-1}(i)} \neq k'_{1p^{\ell-1}(i)}$. Thus, $c_{p^{\ell-1}(i)} = m \oplus k_{1p^{\ell-1}(i)} \neq m \oplus k'_{1p^{\ell-1}(i)} = c'_{p^{\ell-1}(i)}$. □

Lemma 2. *For any* $m \in \{0, 1\}$ *and* $K \in \{0, 1\}^{n-1}$, $\mathsf{conv}(m, K) = \mathsf{conv}(\overline{m}, \overline{K})$.

Proof. The statement immediately follows from the identity $x \oplus y = \bar{x} \oplus \bar{y}$. □

Lemmas 1 and 2 immediately imply the following Theorem 3.

Theorem 3. *For any conversation* $(c_2, c_3, \ldots, c_n)$*, there exists a unique key-value* K *such that* $\mathsf{conv}(0, K) = \mathsf{conv}(1, \overline{K}) = (c_2, c_3, \ldots, c_n)$.

Theorem 3 ensures that our protocol achieves secure message broadcasting: although Eve learns the conversation $(c_2, c_3, \ldots, c_n)$ after our protocol terminates, she cannot obtain any information about whether the message is $m = 0$ or $m = 1$ as implied in Theorem 3.

3. Conclusions

In this paper, we gave a one-round protocol, which achieves secure message broadcasting for any key sharing tree. In other words, we provided a non-interactive secure message broadcasting protocol. Since the previously known protocol, i.e., the flooding protocol, takes h communication rounds where h is the height of a given key sharing tree, our protocol is more efficient than the known one. Furthermore, as seen in Section 2, our protocol is simple. Of course, non-interactivity of our protocol is attractive; note that DC-nets [3,11] have attracted much exploration because of their non-interactivity.

The flooding protocol has been often used as a primitive protocol in multiparty secret key exchange protocols; replacing it with our one-round protocol would bring improvement in such multiparty protocols concerning round complexity.

Our protocol constructed in this paper is oriented for the purpose of message broadcasting rather than multiparty key agreement (multiparty key exchange). Of course, if the designated player P_1 randomly chooses a message m and all players execute our protocol, then the message m can be used as a common secret key; however, in this case, P_1 needs to generate one random bit (namely, a random message m). Alternatively, if the designated player P_1, one of whose children is set to P_i, regards the secret key k_{1i} as a message m and all players execute our protocol, then secret key agreement is achieved without any randomization.

In this paper, we assumed the existence of a public authenticated channel heard by all players. However, in practice, it suffices that the designated player P_1 or each internal player informs only her corresponding descendants (instead of all the players) of her announcements during execution of our protocol. Furthermore, we have considered in this paper all secret keys and messages to be one-bit. One can easily extend our protocol to a multi-bit protocol, provided that all secret keys and messages have the same length, of course.

Acknowledgements

We thank the editor and the anonymous referees whose comments and suggestions helped us to improve the presentation of the paper. This work was supported by KAKENHI (19700002).

References

[1] M. Bläser, A. Jakoby, M. Liśkiewicz, B. Siebert, Private computation — *k*-connected versus 1-connected networks, in: Proc. CRYPTO 2002, in: Lecture Notes in Computer Science, vol. 2442, Springer-Verlag, 2002, pp. 194–209.
[2] M. Burmester, Y. Desmedt, A secure and efficient conference key distribution system, in: Proc. EUROCRYPT '94, in: Lecture Notes in Computer Science, vol. 950, Springer-Verlag, 1995, pp. 275–286.
[3] D. Chaum, The dining cryptographers problem: unconditional sender and recipient untraceability, Journal of Cryptology 1 (1) (1988) 65–75.
[4] Y. Desmedt, Y. Wang, Perfectly secure message transmission revisited, in: Proc. EUROCRYPT 2002, in: Lecture Notes in Computer Science, vol. 2332, Springer-Verlag, 2002, pp. 502–517.
[5] R. Diestel, Graph Theory, second edition, Springer-Verlag, New York, 2000.
[6] M.J. Fischer, R.N. Wright, An application of game-theoretic techniques to cryptography, in: DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 13, AMS, 1993, pp. 99–118.
[7] M.J. Fischer, R.N. Wright, An efficient protocol for unconditionally secure secret key exchange, in: Proc. the 4th Annual Symposium on Discrete Algorithms (SODA), 1993, pp. 475–483.
[8] M.J. Fischer, R.N. Wright, Multiparty secret key exchange using a random deal of cards, in: Proc. CRYPTO '91, in: Lecture Notes in Computer Science, vol. 576, Springer-Verlag, 1992, pp. 141–155.
[9] M. Franklin, M. Yung, Secure hypergraphs: privacy from partial broadcast, SIAM Journal on Discrete Mathematics 18 (3) (2004) 437–450.
[10] M. Franklin, R.N. Wright, Secure communication in minimal connectivity models, Journal of Cryptology 13 (1) (2000) 9–30.
[11] P. Golle, A. Juels, Dining cryptographers revisited, in: Proc. EUROCRYPT 2004, in: Lecture Notes in Computer Science, vol. 3027, Springer-Verlag, 2004, pp. 456–473.
[12] A. Jakoby, M. Liśkiewicz, R. Reischuk, Private computations in networks: topology versus randomness, in: Proc. STACS 2003, in: Lecture Notes in Computer Science, vol. 2607, Springer-Verlag, 2003, pp. 121–132.
[13] T. Mizuki, H. Shizuya, T. Nishizeki, Characterization of optimal key set protocols, Discrete Applied Mathematics 131 (1) (2003) 213–236.
[14] T. Mizuki, H. Shizuya, T. Nishizeki, Dealing necessary and sufficient numbers of cards for sharing a one-bit secret key, in: Proc. EUROCRYPT '99, in: Lecture Notes in Computer Science, vol. 1592, Springer-Verlag, 1999, pp. 389–401.
[15] G.S. Vernam, Cipher printing telegraph systems for secret wire and radio telegraphic communications, Journal of the American Institute for Electrical Engineers 45 (1926) 109–115.
[16] Y. Wang, Y. Desmedt, Secure communication in multicast channels: the answer to Franklin and Wright's question, Journal of Cryptology 14 (2) (2001) 121–135.
[17] C.K. Wong, M. Gouda, S.S. Lam, Secure group communications using key graphs, IEEE/ACM Transactions on Networking 8 (1) (2000) 16–30.
[18] R. Yoshikawa, S. Guo, K. Motegi, Y. Igarashi, Construction of secret key exchange spanning trees by random deals of cards on hierarchical structures, IEICE Trans. Fundamentals E84-A (5) (2001) 1110–1119.