The Tenth International Symposium on Operations Research and Its Applications (ISORA 2011)
Dunhuang, China, August 28–31, 2011

Absolutely Secure Message Transmission Using a Key Sharing Graph

Yoshihiro Indo[1,*] Takaaki Mizuki[2,†] Takao Nishizeki[1,‡]

[1]School of Science and Technology, Kwansei Gakuin University,
2–1 Gakuen, Sanda-shi, 669–1337 Japan
[2]Cyberscience Center, Tohoku University,
Aramaki-Aza-Aoba 6–3, Aoba-ku, Sendai-shi, 980–8578 Japan

Abstract Assume that there are players and an eavesdropper Eve of unlimited computational power and that several pairs of players have shared secret keys beforehand. 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 corresponding to the ends of the edge. Given a key sharing graph, a player wishes to send a message to another player so that the eavesdropper Eve and any other player can get no information on the message. In this paper, we give a necessary and sufficient condition on a key sharing graph for the existence of a protocol for such a purpose.

Keywords Message transmission; Key sharing graph; Eavesdropper; Absolutely secure; 2-connected graph; Internally disjoint paths; Tree

1 Introduction

Consider the following situation. There are $n\,(\geq 2)$ players $p_1, p_2, \cdots, p_n$ and an eavesdropper Eve of unlimited computational power, and several pairs of players have already shared secret keys. For simplicity, each secret key is assumed to be a one-bit number, although the result can be easily generalized to the case where a key can be an ℓ-bit number for any $\ell\,(\geq 1)$. Let G be a graph, called a *key sharing graph* [1, 4, 6, 7], in which each vertex represents a player p_i, $1 \leq i \leq n$, and each edge p_ip_j joining vertices p_i and p_j represents a secret key $k_{ij} \in \{0,1\}$ shared by players p_i and p_j, as illustrated in Fig. 1. (Refer [2] for the graph-theoretic terminology.) A secret key k_{ij} is known only to players p_i and p_j. All players and Eve know the shape of graph G. Every communication (conversation) between players is done through a public communication network and hence can be overheard by any player and Eve, although many conventional models of "secure message transmission" assume a private communication channel [3, 5, 9]. Since Eve has an unlimited computational power, one cannot use any cryptographic protocol based on the computational hardness assumption such as the RSA public key cryptosystem. All players can generate random numbers, and are "honest and faithful," that is, follow a specified protocol and don't "lie."

*yoshihiro.indo7669@kwansei.ac.jp
†tm-paper+isora@g-mail.tohoku-university.jp
‡nishi@kwansei.ac.jp

Under such a situation, if G is a connected graph, that is, G has a spanning tree as drawn by thick lines in Fig. 1, then any player can send a one-bit message to the other players so that the eavesdropper Eve can get no information on the message. Clearly, this can be done, within $n-1$ communication rounds, by the "flooding protocol," which floods the message from p_1 to the other players by using the secret keys on a spanning tree as "one-time pads." This can be done also with exactly one communication round [7].

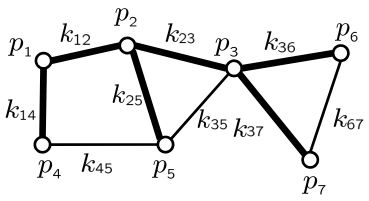

Figure 1: An example of key sharing graph G.

In this paper we consider a protocol by which a player p_1 sends a message $m \in \{0,1\}$ only to a designated player, say p_i, so that Eve and any player other than p_1 and p_i cannot get any information on m. We prove that there is a protocol for such a purpose if and only if G has either an edge p_1p_i joining vertices p_1 and p_i or a pair of paths between p_1 and p_i having no common vertices except p_1 and p_i. This result immediately implies that such a secure message transmission can be done between every pair of players if and only if G is a 2-connected graph.

2 Protocols and main result

In this section we first explain two protocols using one or two paths between p_1 and p_i in a key sharing graph G, and then present our main result.

Let $P = p_{j_0}p_{j_1}\cdots p_{j_\ell}$ be a path between vertices p_1 and p_i in G, where j_0=1 and j_ℓ=i. Player p_1 can send a message $m \in \{0,1\}$ to p_i by the following *protocol I*:

(a) $p_{j_0}(=p_1)$ computes $m \oplus k_{j_0j_1}$ and sends it to p_{j_1}, where $k_{j_0j_1}$ is a secret key shared by p_{j_0} and p_{j_1} and $\oplus$ denotes the logical exclusive OR. Player p_{j_1} gets the message m by computing $m = (m \oplus k_{j_0j_1}) \oplus k_{j_0j_1}$ from $m \oplus k_{j_0j_1}$ and $k_{j_0j_1}$;
(b) a player on P getting the message m sends it to the succeeding player on P, similarly as in (a) above; and
(c) repeat (b) until p_i gets m.

Although this protocol I can transmit the message m from p_1 to p_i, the intermediate players $p_{j_1}, p_{j_2}, \cdots, p_{j_{\ell-1}}$ on path P would know the message m. To the contrast, we say that *p_1 can send a message m to p_i absolutely securely (information-theoretically securely)* if Eve and any player other than p_1 and p_i cannot get any information on m.

Clearly, if there is an edge p_1p_i in G, then p_1 can send a message m to p_i absolutely securely by the protocol I with path $P = p_1p_i$, because there is no intermediate player on

P. Eve and any player other than p_1 and p_i cannot get any information on m, because they don't know the secret key k_{1i}, which is used only once as a "one-time pad" [8].

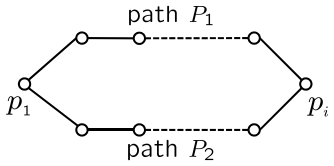

Figure 2: Internally disjoint paths P_1 and P_2 between p_1 and p_i.

Let G have a pair of paths P_1 and P_2 between p_1 and p_i which are *internally disjoint*, that is, have no common vertex except p_1 and p_i, as illustrated in Fig. 2. Then $p_1$can send a message m to p_i absolutely securely by the following *protocol II*. Player p_1 generates a random number $m_1 \in \{0,1\}$ and computes $m_2 = m \oplus m_1$. Player p_1 sends m_1 to p_i by the protocol I above using the secret keys on path P_1, and similarly sends m_2 to p_i using the keys on P_2. Player p_i computes $m = m_1 \oplus m_2$ from m_1 and m_2.

Every intermediate player on path P_1 gets m_1 but cannot get any information on m_2, and hence cannot get any information on m. Similarly, every intermediate player on P_2 cannot get any information on m. Of course, Eve and every player, not on P_1 or P_2, cannot get any information on m.

Thus we have proved the sufficiency in the following theorem.

Theorem 1. A player p_1 can send a message to another player p_i absolutely securely by some protocol if and only if a key sharing graph G has either an edge joining p_1 and p_i or a pair of internally disjoint paths between p_1 and p_i.

We prove the necessity in Theorem 1 in the succeeding section.

A graph G has a pair of internally disjoint paths between p_1 and p_i if and only if a flow network constructed from G has a network flow of value 2 from p_1 to p_i [10]. Therefore, one can examine in linear time whether G has a pair of internally disjoint paths between p_1 and p_i.

A vertex v in a graph G is called a *cut-vertex* if the removal of v from G results in a disconnected graph. The vertex p_3 of the graph in Fig. 1 is a cut-vertex. A graph is defined to be 2-*connected* if there is no cut-vertex. The graph in Fig. 3 is 2-connected. A graph of three or more vertices is 2-connected if and only if there is a pair of internally disjoint paths between any two vertices [2]. Thus, Theorem 1 immediately implies the following corollary.

Corollary 1. Every player can send a message to any other player absolutely securely if and only if a key sharing graph G is 2-connected.

3 Proof of necessity

Before proving the necessity in Theorem 1, we formally define a graph, key information, protocol, communication model, etc. [6].

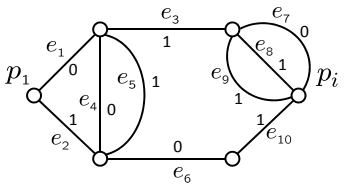

Figure 3: Key sharing graph G, in which a key value is attached to each edge.

We denote by $G = (V,E)$ a *graph* with vertex set V and edge set E. There may exist multiple (parallel) edges in G, as illustrated in Fig. 3. For each vertex $p_v \in V$, we denote by $E(p_v)$ the set of all edges incident to p_v. We denote by $E(p_u, p_v)$ the set of all multiple edges joining vertices p_u and p_v, and hence $E(p_u, p_v) = E(p_u) \cap E(p_v)$.

A vertex p_v in a key sharing graph $G = (V,E)$ represents a player p_v, while an edge e_j in G represents a secret key $k_j \in \{0,1\}$ shared by the two players represented by the ends of e_j. If $|E(p_u, p_v)| = \ell$, then each of the ℓ multiple edges in $E(p_u, p_v)$ represents a 1-bit secret key and hence players p_u and p_v share a secret key of ℓ bits. All the keys are assumed to be uniformly distributed over $\{0,1\}$. Therefore, there are $2^{|E|}$ possibilities. Each can be represented by a set of keys

$$K = \{(e_j, k_j) | e_j \in E\}.$$

Such a set K is called a *key information* of G. There are exactly $2^{|E|}$ key informations, and each of them occurs with probability $1/2^{|E|}$.

For a set $E' \subseteq E$, let

$$K(E') = \{(e_j, k_j) \in K | e_j \in E'\}.$$

For every player $p_v \in V$, the set $K(E(p_v))$ is called a *partial key information* of p_v, and is simply denoted by $K(p_v)$. $K(p_v)$ is known to player p_v.

Let $r_v \in R_v$ be a random number given for each player $p_v \in V$, where R_v is a non-empty finite set. The size of set R_v depends on a protocol and a key sharing graph. For the protocol II in the preceding section, the sender p_1 needs a one-bit random number m_1, and hence $|R_1| = 2$. On the other hand, every other player p_v needs no random number, and hence we set $|R_v| = 1$. The set $\mathscr{R} = \{(p_v, r_v) | p_v \in V\}$ is called a *random number set* of G.

In our protocol it is assumed that each player $p_v \in V$ is given a partial key information $K(p_v)$ and a random number $r_v \in R_v$ as her input. In the first communication round of a protocol, the sender p_1 broadcasts a cryptogram[1], computed from a message m, a partial key information $K(p_1)$ and a random number r_1, to all players through a public communication network, while every other player $p_v \in V - \{p_1\}$ broadcasts a cryptogram computed only from $K(p_v)$ and r_v. In the succeeding communication rounds, every player p_v broadcasts a cryptogram computed from $K(p_v)$, r_v and all the cryptograms that have

[1]Precisely speaking, it is not necessarily a cryptogram, but we call it a cryptogram for convenience.

been broadcasted through a network. Finally, the receiver p_i computes a value in $\{0,1\}$ from $K(p_i)$, r_i and all the broadcasted cryptograms, and output it as a message received from p_1. We say that p_1 *can send a message m to* p_i if the value output by p_i is equal to the message m. The set of all cryptograms that are broadcasted during the execution of a protocol is called a *conversation* σ. Eve and every player can overhear σ since all cryptograms are broadcasted through a public communication network.

The triplet $(K,\mathscr{R},m)$ of a key information K, a random number set $\mathscr{R}$ and a message m is called a *protocol point*. Given a protocol and a key sharing graph G with designated *sender* p_1 and *receiver* p_i, both the conversation σ and the value output by p_i are uniquely decided only by a protocol point $(K,\mathscr{R},m)$. Player p_1 must send a message to p_i absolutely securely for any protocol point $(K,\mathscr{R},m)$. Therefore, neither Eve nor any player other than p_1 and p_i can get any information on a message m. Hence, for any probability distribution $P(m)$ of message $m \in \{0,1\}$, any conversation σ and any player p_v other than p_1 and p_i,

$$P(m|\sigma) = P(m) \tag{1}$$

and

$$P(m|\sigma,K(p_v),r_v) = P(m), \tag{2}$$

where $P(m|\sigma)$ is the (conditional) probability of a message being m under the condition that a conversation is σ, and $P(m|\sigma,K(p_v),r_v)$ is similarly defined. (Obviously, Eq. (2) implies Eq. (1) if there are three or more players.)

We have thus formally defined a protocol, a communication model, etc, and are now ready to prove the necessity in Theorem 1.

Proof of the necessity in Theorem 1

Suppose for a contradiction that p_1 can send any message m to p_i absolutely securely by some protocol although a key sharing graph G has neither edge p_1p_i nor a pair of internally disjoint paths between p_1 and p_i. Then we can derive a contradiction, as follows.

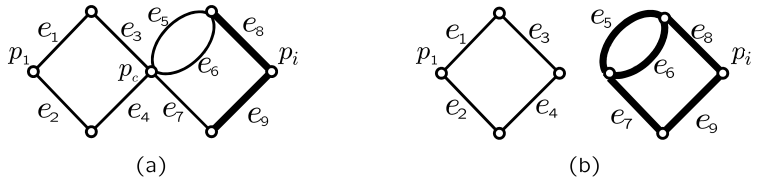

Figure 4: Graph G and subgraph G_i.

We first define a subgraph $G_i = (V_i, E_i)$ of G. If vertices p_1 and p_i are contained in the same connected component of G as illustrated in Fig. 4(a), then there is a cut-vertex p_c such that p_1 and p_i are contained in different connected components in the graph $G - p_c$ obtained from G by deleting p_c, because G has no pair of internally disjoint paths

between p_1 and p_i. Let $G_i = (V_i, E_i)$ be the connected component of $G - p_c$ containing p_i. The subgraph G_i is drawn by thick lines in Fig. 4(a). On the other hand, if p_1 and p_i are contained in different connected components of G as illustrated in Fig. 4(b), then there is no path between p_1 and p_i and let $G_i = (V_i, E_i)$ be the connected component of G containing p_i. In Fig. 4(b) G_i is drawn by thick lines.

Let σ be the conversation for a protocol point $(K, \mathscr{R}, 1)$. We write $K = \{(e_j, k_j) | e_j \in E\}$ and $\mathscr{R} = \{(p_v, r_v) | p_v \in V\}$, as illustrated in Fig. 5. Since $m = 1$, p_i outputs 1.

There is a protocol point $(K', \mathscr{R}', 0)$ with the same conversation σ above. Otherwise, a message is always 1 under the condition that a conversation is σ, and hence,

$$P(m|\sigma) = \begin{cases} 1 & \text{if } m = 1; \\ 0 & \text{otherwise,} \end{cases}$$

contradicting to the assumption that Eq. (1) holds for any probability distribution $P(m)$ of message m. Let $K' = \{(e_j, k'_j) | e_j \in E\}$ and $\mathscr{R}' = \{(p_v, r'_v) | p_v \in V\}$, as illustrated in Fig. 6. Of course, p_i outputs 0 for the protocol point $(K', \mathscr{R}', 0)$.

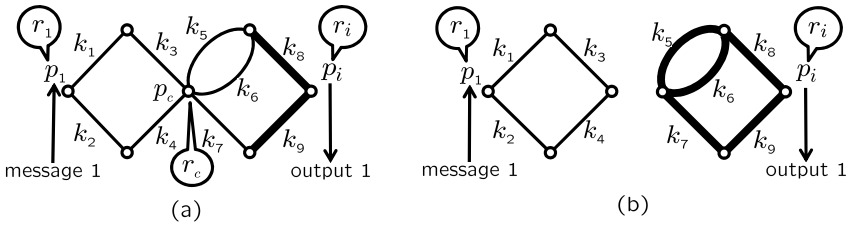

Figure 5: Protocol point $(K, \mathscr{R}, 1)$.

In particular, if p_1 and p_i are contained in the same connected component, then there is a protocol point $(K', \mathscr{R}', 0)$ with conversation σ such that $K'(p_c) = K(p_c)$ and $r'_c = r_c$. Otherwise,

$$P(m|\sigma, K(p_c), r_c) = \begin{cases} 1 & \text{if } m = 1; \\ 0 & \text{otherwise,} \end{cases}$$

contrary to Eq. (2).

We then consider a protocol point $(K'', \mathscr{R}'', 1)$ constructed from $(K, \mathscr{R}, 1)$, $(K', \mathscr{R}', 0)$ and $G_i = (V_i, E_i)$ as follows:

$$K'' = K(E - E_i) \cup K'(E_i) \tag{3}$$

and

$$\mathscr{R}'' = \{(p_v, r_v) \in \mathscr{R} | p_v \in V - V_i\} \cup \{(p_v, r'_v) \in \mathscr{R}' | p_v \in V_i\}. \tag{4}$$

Figure 7 illustrates $(K'', \mathscr{R}'', 1)$ for $(K, \mathscr{R}, 1)$ in Fig. 5 and $(K', \mathscr{R}', 0)$ in Fig. 6. For the protocol point $(K'', \mathscr{R}'', 1)$, p_i must output 1.

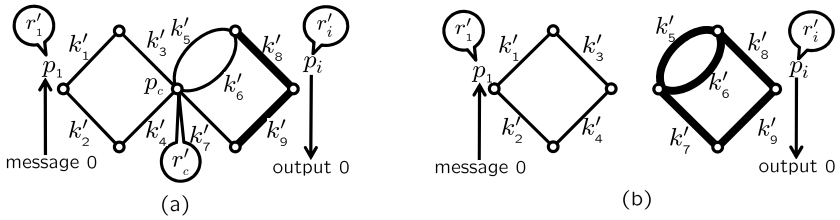

Figure 6: Protocol point $(K', \mathscr{R}', 0)$.

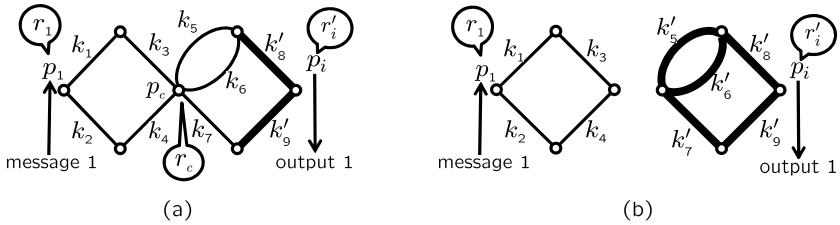

Figure 7: Protocol point $(K'', \mathscr{R}'', 1)$ obtained from $(K, \mathscr{R}, 1)$ and $(K', \mathscr{R}', 0)$.

Let $\mathscr{R}'' = \{(p_v, r''_v) | p_v \in V\}$. Then we now claim that

$$K''(p_v) = K(p_v) \text{ and } r''_v = r_v \tag{5}$$

for every $p_v \in V - V_i$ and

$$K''(p_v) = K'(p_v) \text{ and } r''_v = r'_v \tag{6}$$

for every $p_v \in V_i$. Cleary, the claim holds if p_1 and p_i are contained in different connected components of G. Furthermore, the claim holds for p_v if either $p_v \in V - V_i$ or p_v is not adjacent to p_c in G. Therefore, one may assume that p_1 and p_i are contained in the same connected component of G, $p_v \in V_i$, and p_v is adjacent to p_c in G. Then $E(p_v) = E(p_v, p_c) \cup (E(p_v) \cap E_i)$ and $E(p_v, p_c) \subseteq E - E_i$. Since $K'(p_c) = K(p_c)$ and $E(p_v, p_c) \subseteq E(p_c)$, we have $K'(E(p_v, p_c)) = K(E(p_v, p_c))$. Therefore, by Eq. (3) we have

$$\begin{aligned} K''(p_v) &= K(E(p_v, p_c)) \cup K'(E(p_v) \cap E_i) \\ &= K'(E(p_v, p_c)) \cup K'(E(p_v) \cap E_i) \\ &= K'(E(p_v)). \end{aligned}$$

Since $p_v \in V_i$, by Eq. (4) we have $r''_v = r'_v$. Thus Eq. (6) holds for p_v.

We now claim that the conversation for $(K'', \mathscr{R}'', 1)$ is the same as σ. Since $K''(p_1) = K(p_1)$, $r''_1 = r_1$ and $m = 1$, the cryptogram broadcasted by p_1 for $(K'', \mathscr{R}'', 1)$ is the same as that for $(K, \mathscr{R}, 1)$. A cryptogram broadcasted by any player other than p_1 for $(K'', \mathscr{R}'', 1)$ is the same as either that for $(K, \mathscr{R}, 1)$ or that for $(K', \mathscr{R}', 0)$, because either $K''(p_v) =$

$K(p_v)$ and $r''_v = r_v$ or $K''(p_v) = K'(p_v)$ and $r''_v = r'_v$ and both the conversation for $(K, \mathscr{R}, 1)$ and that for $(K', \mathscr{R}', 0)$ are σ.

However, p_i would output the value 0 for $(K'', \mathscr{R}'', 1)$, because $K''(p_i) = K'(p_i)$, $r''_i = r'_i$, and p_i outputs 0 for $(K', \mathscr{R}', 0)$. This is a contradiction, because p_i outputs 1 for $(K'', \mathscr{R}'', 1)$. □

4 Conclusions

We showed that a player p_1 can send a message to another player p_i absolutely securely if and only if a key sharing graph G has either an edge p_1p_i or a pair of internally disjoint paths between p_1 and p_i. This immediately implies that every player can send a message to any other player absolutely securely if and only if G is 2-connected. For simplicity, we assumed that a message and all secret keys are 1-bit numbers. However, one can easily extend the results to the case where a message and all secret keys are ℓ-bit numbers for any $\ell(\geq 1)$. In this case, $\oplus$ must be a bit-wise exclusive OR.

We obtained also a sufficient condition for a player p_1 to be able to send a message to a multiple number of designated players absolutely securely. However, the detail is omitted in this extended abstract.

References

[1] D. Chaum, "The dining cryptographers problem: unconditional sender and recipient untraceability," Journal of Cryptology, 1(1), pp.65-75, 1988.

[2] R. Diestel, Graph Theory, second edition, Springer-Verlag, New York, 2000.

[3] D. Dolve, C. Dwork, O. Waarts and M. Young, "Perfectly secure message transmission," Journal of ACM, 1(40), pp.17-47, 1993.

[4] M. J. Fisher and R. N. Wright, "Multiparty secret key exchange using a random deal of cards," Proc. CRYPTO '91, Lecture Notes in Computer Science, 576, Springer-Verlag, pp.141-155, 1992.

[5] K. Kurosawa and K. Suzuki, "Truly efficient 2-round perfectly secure message transmission scheme," In Advances in Cryptology-EUROCRYPT 2008, Lecture Notes in Computer Science 4965, pp.324-340, Springer, 2008.

[6] T. Mizuki, S. Nakayama and H. Sone, "An application of *st*-numbering to secret key agreement," Int. J. Foundations of Computer Science, to appear.

[7] T. Mizuki, T. Sato and H. Sone, "A one-round secure message broadcasting protocol through a key sharing tree," Inf. Proc. Let., 109, pp.842-845, 2009.

[8] C. Shannon, "Communication theory of secrecy systems," Bell System Technical Journal, 28, pp.656-715, 1949.

[9] K. Srinathan, A. Narayanan and C. Pandu Rangan, "Optimal perfectly secure message transmission," In Advances in Cryptology-CRYPTO'04, Lecture Notes in Computer Science 3152, pp.545-561, Springer-Verlag, 2004.

[10] H. Suzuki, T. Akama and T. Nishizeki, "Algorithms for finding internally disjoint paths in a planar graph," IEICE Trans. A, J71-A(10), pp.1906-1916, 1988.