Cryptanalysis of Secure Message Transmission Protocols with Feedback

(Full version of the ICITS 2009 paper with the same title and authors)

Qiushi Yang[1,⋆] and Yvo Desmedt[1,2,⋆⋆]

[1] Department of Computer Science, University College London, UK
{q.yang, y.desmedt}@cs.ucl.ac.uk
[2] RCIS, AIST, Japan

Abstract. In the context of secure point-to-point message transmission in networks with minimal connectivity, previous studies showed that feedbacks from the receiver to the sender can be used to reduce the requirements of network connectivity. We observe that the way how feedbacks were used in previous work does not guarantee perfect privacy to the transmitted message, when the adversary performs a *Guessing Attack*. In this paper, we shall describe our new Guessing Attack to some existing protocols (in fact, we are the first to point out a flaw in the protocols of Desmedt-Wang's Eurocrypt'02 paper and of Patra-Shankar-Choudhary-Srinathan-Rangan's CANS'07 paper), and propose a scheme defending against a general adversary structure. In addition, we also show how to achieve almost perfectly secure message transmission with feedbacks when perfect reliability or perfect privacy is not strictly required.

Keywords: secure message transmission, privacy and reliability, Guessing Attack, adversary structure, feedback.

1 Introduction

Secure point-to-point communication requires both private and reliable message transmission from a sender A to a receiver B, despite the possibility that some parties on the channels between them are corrupted. Dolev et al. [8] initialized the problem of secure message transmission by showing that secure communication is possible in a network graph that is not complete. The interplay of the network connectivity and secure communication has been studied extensively [7, 11, 2, 4, 8, 9, 5, 14, 6, 26, 24, 15].

The general setting of this problem assumes an active *Byzantine* adversary, who has unlimited computational power (not only a passive listener). An adversary X can be characterized as *threshold* (k-bounded) or *non-threshold* (general adversary structure). In the initial studies, Dolev [7] and Dolev et al. [8] showed that $2k+1$ connectivity is required for reliable message transmission, and if all communication links are one-way, then the system's network needs to be $3k+1$ connected. Some further studies on threshold adversaries have been done by Franklin and Wright [9], Srinathan and Rangan [26], Shankar et al. [24], and Kurozawa and Suzuki [15]. Furthermore, in the presence of a *general adversary structure* [12], Kumar et al. [14] gave the necessary and sufficient conditions for perfectly secure message transmission in bi-direction networks (all links are two-way), and later, Desmedt et al. [6] extended the research and provided some results on all-one-way linked networks.

Although the concerning problem may seem trivial, it is far from straightforward. Many solutions on the topic of secure message transmission require careful examination. For instance, in

⋆ Funded by a UCL PhD studentship.
⋆⋆ This work was done while funded by EPSRC EP/C538285/1 and by BT, as BT Chair of Information Security.

Crypto 04, Srinathan et al. [25] proposed an optimal (in transmission rate) protocol for all-two-way communication. However, that protocol was later proved not perfectly reliable as originally claimed, by Agarwal et al. [1]. Their work appeared in Crypto 06. Similarly, in this work, we show that perfect privacy can be breached in many schemes that use the so-called *feedback channels* (e.g. some protocols of Desmedt and Wang [5] in Eurocrypt'02).

Given a sender A and a receiver B in a network. The channel that A uses to transmit a message to B is called the *forward channel*, and the channel that B transmits feedbacks to A is called the *feedback channel*. In an all-two-way linked network, the forward channels and the feedback channels have the same connectivity (symmetric). That is, if B can reliably receive message from A, then A can reliably receive feedbacks from B. However, this is only a special case of using feedbacks. In general, the feedbacks that A receives may not be reliable. That is, the feedback channels may have less connectivity than the forward channels do. Desmedt and Wang [5] motivated this with the following scenarios: a channel from A to B is cheap, but a channel from B to A is expensive; in another scenario, A has access to more resources than B does.

Some studies have been done concerning this network setting (with unreliable feedback channels). This problem was initialized by Desmedt and Wang [5] in Eurocrypt'02. In their paper they showed that if there are u directed node-disjoint paths from B to A, then it is sufficient to have $3k+1-u > 2k+1$ directed node-disjoint paths from A to B against a k-active adversary. Another study has been done by Patra et al. [20], in which they extended the previous results and considered a general adversary structure.[3] However, we observe that *all* the protocols in these papers are not so *perfectly* secure as they claimed, as those protocols actually leak some information about the message to the adversary X, when X corrupts the feedback channel and acts on it. Thus we shall show how X can attack those protocols in this paper.

Our contributions. In our work we study the use of the feedback channels in depth. Particularly, we observe that the major functionality of the feedback channels is to be used by the receiver B for reliable message transmission purpose when faulty messages are received, but this may undermine perfect privacy of the transmitted messages. We will describe a new *Guessing Attack* that the adversary may perform on many existing protocols that work in networks with feedback channels.

Next we show how to construct a perfectly secure message transmission protocol that withstands the Guessing Attack and any other attack. In this paper we consider a general adversary structure, thus our results can be applied in more general cases. In addition, we study *almost* perfectly secure message transmission. First we show that the network connectivity required for achieving almost perfectly private message transmission is *exactly the same* as that for achieving perfect privacy. Next, we study almost perfectly reliable message transmission tolerating a general adversary structure, and propose a protocol, which is a generalization of the result in [5].

Organization of this paper. We describe our model in Section 2. In Section 3 we propose our Guessing Attack that breaches perfect privacy of some existing protocols. Section 4 is devoted to present the necessary and sufficient conditions for perfectly secure message transmission, and we shall give our main protocol that tolerates the Guessing Attack in this section. In Section 5, we show our result on almost perfectly private message transmission, and in Section 6, we discuss almost perfectly reliable message transmission.

[3] We noticed that some recent studies have been done considering this network setting (see [18, 19]). However, those results are less relative to our concern.

2 Model and background

Basic definitions. We abstract away the concrete network structure and model a network by a directed graph $G(V, E)$, whose nodes are the parties in the network and edges are point-to-point secure communication links, where all the edges in E have directions. We also denote $\mathbb{F}$ as the finite field that both A and B agree on, and $\mathcal{M} \subseteq \mathbb{F}$ as the message space that A chooses message from. Let S be a set, we write $|S|$ to denote the number of elements in S, and $a \in_R S$ to indicate that a is chosen from S with respect to the uniform distribution. Let $a \in \mathbb{R}$. We write $\lfloor a \rfloor \in \mathbb{Z}$ to denote the integer part of a. Let $a, b, M \in \mathbb{F}$. We employ an authentication function $auth(M; a, b) := aM + b$, by which each authentication key $key = (a, b)$ can be used to authenticate one message M without revealing any information about the authentication key (see [10, 22, 21, 9]).

Throughout the paper, we assume that $A, B \in V$, and use $\mathcal{P}$ as the set of all the directed paths from A to B and $\mathcal{Q}$ as the set of all the directed paths from B to A (the directed paths are not necessarily node-disjoint). Let $Z \subseteq V$, we write $\mathcal{P}_Z$ to denote the set of all paths in $\mathcal{P}$ that pass through nodes in Z, and write $\bar{\mathcal{P}}_Z$ to denote the set of all paths in $\mathcal{P}$ that are free of nodes in Z. Similarly, we denote $\mathcal{Q}_Z$ and $\bar{\mathcal{Q}}_Z$.

Secret sharing. We define a $(k+1)$-out-of-n ϵ-private secret sharing scheme ($(k+1, n, \epsilon)$-SSS).

Definition 1 *Let $\epsilon < 1$. A $(k+1, n, \epsilon)$-SSS is a probabilistic function $S : \mathbb{F} \to \mathbb{F}^n$ such that for any $m \in \mathbb{F}$ and $(v_1, ..., v_n) = S(m)$,*
property-1 m can be recovered from any $k+1$ entries of $(v_1, ..., v_n)$ with probability 1, and
property-2 m can also be recovered from any $r \leq k$ entries with probability at most ϵ.

Therefore, the classic Shamir's scheme [23] is a $(k+1, n, 0)$-SSS, and Blakely's scheme [3] is a $(k+1, n, \epsilon)$-SSS (almost perfectly private). The set of all possible $(v_1, ..., v_n)$ can be viewed as a code and its elements codewords. When there is no ambuiguity, we view $S(m)$ as a subset of this code. We say a $(k+1, n, \epsilon)$-SSS can detect d errors if given any codeword $(v_1, ..., v_n)$ and any tuple $(u_1, ..., u_n)$ such that $0 < |i : u_i \neq v_i, 1 \leq i \leq n| \leq d$, one can detect that $(u_1, ..., u_n)$ is not a codeword; a $(k+1, n, \epsilon)$-SSS can correct c errors if given $(v_1, ..., v_n) \in S(m)$, from any tuple $(u_1, ..., u_n)$ such that $|i : u_i \neq v_i, 1 \leq i \leq n| \leq c$, one can recover the secret m. It has been proved that a $(k+1, n, 0)$-SSS can detect $n-k-1$ errors and correct (not simultaneously) $\lfloor (n-k-1)/2 \rfloor$ errors using error-correcting code [16, 17].

Adversary model. We consider an adversary X who is characterized by an adversary structure $\mathcal{Z}$ that consists of all sets of parties that X can corrupt. A definition of an adversary structure was given by Hirt and Maurer [12] (see also [13]): Given a party set P, an adversary structure $\mathcal{Z}$ on P is a family of subsets $\mathcal{Z} \subset 2^P$ such that: $Z \in \mathcal{Z}, Z' \subseteq Z \subseteq P \Rightarrow Z' \in \mathcal{Z}$. A set $Z \in \mathcal{Z}$ is called *maximal* if $Z' \supset Z \Rightarrow Z' \notin \mathcal{Z}$, and we use $\tilde{\mathcal{Z}}$ as the set of all maximal sets in $\mathcal{Z}$.

Throughout the paper we use $Z_x \in \mathcal{Z}$ to denote the set of parties that the adversary X chooses to control. We allow an *active*, or *Byzantine*, adversary, who has unlimited computational power and resources. The adversary X can read the traffic of Z_x and perform any local computation on Z_x. In this paper we only consider a static adversary, whose choice of Z_x does not change throughout the protocol.

Message transmission protocol. Let Π be a message transmission protocol. A starts with a message M^A drawn from a message space $\mathcal{M}$ with respect to a certain probability distribution. At the end of the protocol Π, B outputs a message $M^B \in \mathcal{M}$. For any execution of the protocol Π, let adv

be the adversary X's view of the entire protocol. We write $adv(M, r)$ to denote X's view when $M^A = M$ and when the sequence of coin flips used by X is r (follows [9, 6]).

Privacy: Π is ϵ-*private* if, for any two messages $M_0, M_1 \in \mathcal{M}$ and every r, $\sum_c |Pr[adv(M_0, r) = c] - Pr[adv(M_1, r) = c]| \leq 2\epsilon$.

Reliability: Π is δ-*reliable* if, with probability at least $1-\delta$ ($0 \leq \delta < \frac{1}{2}$), B terminates $M^B = M^A$.

Security: Π is (ϵ, δ)-secure if it is ϵ-private and δ-reliable.

We say Π is a perfectly secure message transmission protocol if it is $(0, 0)$-secure. In this paper, we also discuss $(0, \delta)$-secure and $(\epsilon, 0)$-secure message transmissions, which are almost perfectly secure.

In the presence of an adversary structure $\mathcal{Z}$, Kumar et al. [14] showed that in a bi-direction network, the necessary and sufficient condition for $(0, 0)$-secure message transmission from A to B is that $\mathcal{P}_{Z_a \cup Z_b} \subsetneq \mathcal{P}$ for any $Z_a, Z_b \in \mathcal{Z}$. In the case that all communication links are one-way without feedback, Desmedt et al. [6] proved that 0-reliable message transmission from A to B can be achieved if and only if $\mathcal{P}_{Z_a \cup Z_b} \subsetneq \mathcal{P}$ for any $Z_a, Z_b \in \mathcal{Z}$, and $(0, 0)$-secure message transmission is possible if and only if $\mathcal{P}_{Z_a \cup Z_b \cup Z_c} \subsetneq \mathcal{P}$ for any $Z_a, Z_b, Z_c \in \mathcal{Z}$. Furthermore, we will discuss the case, in which the feedback channels exist, in Section 4.

3 Attack on feedback channels

In this section we propose a *Guessing Attack* that takes advantage of how the feedback channels are normally used. In most protocols that work on networks with feedback channels, the feedbacks are used by the receiver B to seek for help from A when B does not have enough information to recover the message (i.e., for reliability purpose). In our attack, we propose the following. Since the adversary X can choose to corrupt some feedback paths, it can simulate how B uses the feedback channels and learn from A the information it needs to recover the message with better probability than guessing. This allows X to breach perfect privacy, as we describe now in more detail.

Here we give an example of how Guessing Attack breaches perfect privacy of one of Desmedt and Wang's protocols in [5]. This DW protocol (the protocol corresponding to [5, Theorem 5]) is for $(0, 0)$-secure message transmission against a threshold adversary. First we shall sketch the DW protocol before we show that it is not 0-private.

Condition for the DW protocol: there are $3k \geq 2k + 1$ directed node-disjoint paths from A to B and one directed node-disjoint path from B to A.[4]

Sketch of the DW protocol Let $p_1, ..., p_{3k}$ be the directed paths from A to B and q be the directed path from B to A.

Step 1 ...

Step 2 A chooses a $key^A \in_R \mathbb{F}$ and constructs $(k + 1, 3k, 0)$-secret-shares $v = (s_1, ..., s_{3k})$ of key^A. For each $1 \leq i \leq 3k$, A sends s_i to B via path p_i.

Step 3 Let $v^B = (s_1^B, ..., s_{3k}^B)$ be the shares B receives. If B finds that there are at most $k - 1$ errors (using error-correcting code), B recovers key^B from the shares, sends 'stop' to A via path q; otherwise, B sends v^B to A via path q.

Step 4 If A receives $v^A = (s_1^A, ..., s_{3k}^A)$ from path q, A broadcasts $P = \{i : s_i^A \neq s_i\}$ ($|P| = k$) via all paths $p_1, ..., p_{3k}$; otherwise, A broadcasts 'stop'.

[4] This condition is sufficient for $(0, 0)$-secure message transmission from A to B, but is stronger than the necessary condition. See [5] for more detail.

The k-active adversary X chooses to control paths $p_1, ..., p_{k-1}$ and path q. Thus X is able to get shares $(s_1, ..., s_{k-1})$ in Step 2. With these $k-1$ shares, X performs the following:
X chooses a share $s_k^X \in_R \mathbb{F}$ and two keys $key_1^X, key_2^X \in_R \mathbb{F}$ ($key_1^X \neq key_2^X$). Corresponding to key_1^X, X assumes that $(s_1, ..., s_{k-1}, s_k^X)$ are k shares of key_1^X, thus using Lagrange interpolation, X gets another k shares $(s_{k+1}^X, ..., s_{2k}^X)$ of key_1^X. Similarly, corresponding to key_2^X, X assumes that $(s_1, ..., s_{k-1}, s_k^X)$ are k shares of key_2^X, and gets another k shares $(s_{2k+1}^X, ..., s_{3k}^X)$ of key_2^X. X sets $v^X = (s_1, ..., s_{k-1}, s_k^X, ..., s_{3k}^X)$.
In each execution step of the DW protocol, X acts passive on paths $p_1, ..., p_{k-1}$. Thus B sends 'stop' to A in Step 3. On the feedback path q that X corrupts, X ignores what B sends and forwards v^X to A. Then in Step 4, if A finds exactly k errors in $v^A = v^X$, A broadcasts $P = \{i : s_i^X \neq s_i\}$, according to which X recovers $key^A = key_j^X$ ($j \in \{1, 2\}$); otherwise, A broadcasts 'stop', and X randomly guesses a key^X.

Fig. 1. Guessing Attack to the DW protocol.

Step 5 ...
Step 6 A broadcasts $key^A + M^A$ via all paths $p_1, ..., p_{3k}$, where M^A is the actual message.
Step 7 ...

This single feedback channel protocol is the basis of the main protocols in [5]. We observe that this DW protocol is 0-reliable, so in the above sketch we did not describe how B recovers the message (see [5] for the entire protocol). Now we show that using our Guessing Attack, the adversary X can learn the message M^A with probability better than guessing.

Theorem 1 *This DW protocol is not a* 0*-private message transmission protocol from* A *to* B*.*

Proof. Due to the fact that $key^A \in_R \mathbb{F}$, if this DW protocol is 0-private, then the probability that the adversary X guesses key^A is $\frac{1}{|\mathbb{F}|}$. That is, X learns nothing from the shares it gets, and can only guess a uniformly random number $key^X \in \mathbb{F}$, and with probability $\frac{1}{|\mathbb{F}|}$, $key^X = key^A$. We call this a *random guess*. Now we show a Guessing Attack by which X can learn key^A with a probability better than $\frac{1}{|\mathbb{F}|}$ (see Fig.1).

In this Guessing Attack, X guesses a share s_k^X and two keys key_1^X and key_2^X. It is straightforward that A will broadcast P if and only if A finds exactly k errors in v^X, and the k errors can only be either $(s_{k+1}^X, ..., s_{2k}^X)$ or $(s_{2k+1}^X, ..., s_{3k}^X)$. That is, the guess is successful if $s_k^X = s_k$ and one of the two keys is correct (i.e., $key_i^X = key^A, i \in \{1, 2\}$). Thus the probability T that the guess is successful is

$$T = \frac{1}{|\mathbb{F}|} \times \left(2 \times \frac{1}{|\mathbb{F}|}\right) = \frac{2}{|\mathbb{F}|^2}.$$

If the guess fails, then X will use a random guess with probability $\frac{1}{|\mathbb{F}|}$ to get $key^X = key^A$. Thus, the total probability G that X learns key^A by performing Guessing Attack is

$$G = T + (1 - T) \times \frac{1}{|\mathbb{F}|} > \frac{1}{|\mathbb{F}|}.$$

Therefore, X can learn key^A with a probability better than $\frac{1}{|\mathbb{F}|}$ and simultaneously recover M^A with probability better than guessing.[5] Hence we proved that the DW protocol is not 0-private. □

[5] Although M^A can be chosen with respect to any probability distribution (not necessarily uniform), more knowledge of the key key^A gives better probability of getting M^A.

Note that in journal paper [27], Wang and Desmedt provided a new protocol that uses induction when A receives tuples of shares in feedbacks (the case that Guessing Attack may happen). When A notices that Guessing Attack may happen according to the feedbacks it receives, it uses an induction and re-sends the message without revealing the message to the adversary (0-private). The property of the threshold adversray, t-bounded, allows the induction to be continued until the message is transmitted 0-reliably. Thus the protocol in [27] enables perfect security. For details of the $(0,0)$-secure message transmission protocol tolerating a threshold adversary, we refer to [27, Theorem 4.2].

As we showed in the above example, the basic idea of Guessing Attack is to replace the feedbacks from B to A on the feedback channel with something that may reveal the message. There is some probability associated with this guessing of being successful.

Besides the Desmedt-Wang protocols, we observe that all protocols given by Patra et al. in [20] that tolerate either threshold or non-threshold adversaries do not guarantee perfect privacy when the Guessing Attack takes place, and hence they are not $(0,0)$-secure. We show our Guessing Attacks to Protocol I, Protocol II and Secure Protocol from [20] in Appendix A.

4 $(0,0)$-secure message transmission

In this section, we address the question of perfectly secure message transmission, for which both 0-private and 0-reliable message transmissions are required. That is, we shall provide a new protocol that tolerates the Guessing Attack. We focus on a $(0,0)$-secure message transmission against a general adversary structure (as Wang and Desmedt [27] recently provided a $(0,0)$-secure protocol for the threshold case), hence our protocol can be used in more general cases. Before we show our protocol, we generalize the following theorem based on the result by Patra et al. [20].

Theorem 2 *Let $G(V,E)$ be a directed graph, $\mathcal{Z}$ be an adversary structure on $V \setminus \{A,B\}$, and $\mathcal{Q} \neq \emptyset$. The necessary and sufficient conditions (CONs) for $(0,0)$-secure message transmission from A to B are:*

CON-1 for any two sets $Z_a, Z_b \in \mathcal{Z}$: $\mathcal{P}_{Z_a \cup Z_b} \subsetneq \mathcal{P}$, and

CON-2 for any three sets $Z_a, Z_b, Z_c \in \mathcal{Z}$, if $\mathcal{P}_{Z_a \cup Z_b \cup Z_c} = \mathcal{P}$, then out of the three sets, there is at most one Z_i ($i \in \{a,b,c\}$) such that $\mathcal{Q}_{Z_i} = \mathcal{Q}$.

We also employ a lemma from [20] for a simpler protocol, as using this lemma, we only need to consider a set $\tilde{\mathcal{Y}}$ of size 3 that contains the set $Z_x \in \tilde{\mathcal{Z}}$ that the adversary X chooses to control.

Lemma 1 (see [20]) *Let $\mathcal{Z}$ be an adversary structure on $V \setminus \{A,B\}$. $(0,0)$-secure message transmission from A to B tolerating $\mathcal{Z}$ is possible if: for any monotone subset $\mathcal{Y} \subseteq \mathcal{Z}$ such that $|\tilde{\mathcal{Y}}| = 3$ and $Z_x \in \tilde{\mathcal{Y}}$, there is a $(0,0)$-secure message transmission protocol from A to B tolerating $\tilde{\mathcal{Y}}$.*

In [20], Patra et al. proposed a *Secure Protocol* tolerating $\tilde{\mathcal{Y}}$. However, the Secure Protocol is vulnerable to Guessing Attack, and hence is not 0-private (see Appendix A for the proof).

Now we show a $(0,0)$-secure message transmission protocol (PSP) under CONs tolerating such a sub-structure $\tilde{\mathcal{Y}}$ and defending Guessing Attack. First we let $\tilde{\mathcal{Y}} = \{Z_1, Z_2, Z_3\}$. The case that $\mathcal{P}_{Z_1 \cup Z_2 \cup Z_3} \subsetneq \mathcal{P}$ has been proved in [6]. Now we consider the case that $\mathcal{P}_{Z_1 \cup Z_2 \cup Z_3} = \mathcal{P}$. Here we employ the similar settings to the proof to [20, Theorem 10]; that is, due to CON-1, three forward paths $p_1 \in \bar{\mathcal{P}}_{Z_2 \cup Z_3}$, $p_2 \in \bar{\mathcal{P}}_{Z_1 \cup Z_3}$ and $p_3 \in \bar{\mathcal{P}}_{Z_1 \cup Z_2}$ exist to transmit messages from A to B. This

implies that, since $Z_x \in \tilde{\mathcal{Y}}$, the adversary X can corrupt at most one p_i $(1 \leq i \leq 3)$. Thus if A sends a value via all three paths p_1, p_2, p_3, then B can recover this value using a majority vote. In our protocol we say that A *reliably sends* a value to B to indicate this kind of transmission.

Based on CON-2, we assume that $\mathcal{Q}_{Z_1} \subsetneq \mathcal{Q}$, $\mathcal{Q}_{Z_2} \subsetneq \mathcal{Q}$ and $\mathcal{Q}_{Z_3} \subseteq \mathcal{Q}$. Moreover, due to CON-2, two feedback paths $q_1 \in \bar{\mathcal{Q}}_{Z_1}$ and $q_2 \in \bar{\mathcal{Q}}_{Z_2}$ exist to transmit feedbacks from B to A.

In our protocol, we use 0 as default received value. That is, when A is sending to B, if B receives nothing on path $p \in \mathcal{P}$, then B assumes that 0 is received on path p. Similarly if A receives nothing on path $q \in \mathcal{Q}$ from B, then A assumes that 0 is received on path q.

Underlying idea. Our protocol runs a loop. In each round of the loop, the feedback paths q_1 and q_2 are used to transmit only one bit: either 0 or 1. This prevents the Guessing Attack from happening at the first place. If in a round of the loop, B found that one of the forward paths p_1, p_2 or p_3 transmits a faulty message, then B will send 0 via the feedback paths. If A receives 0 on q_j $(j \in \{1, 2\})$, then A will reliably send the message to B again, so B will then know which path p_f $(1 \leq f \leq 3)$ is faulty. In the rest of the protocol, B will only recover the message on p_i and p_j $(i, j \in \{1, 2, 3\} \setminus \{f\})$, and will not send 0 as feedback again. Therefore, if A receive 0 on q_j $(j \in \{1, 2\})$ more than once, then A knows that q_j is faulty, and will not consider the feedbacks received on q_j again in the rest of the protocol. In our protocol, we let A use err_1 and err_2 to count the numbers of 0's received on paths q_1 and q_2 respectively. Furthermore, if in a round of the loop, A does not receive 0 on the feedback path(s) that A considers not faulty, then A will not send any information about the message again, and A knows that the message has been transmitted 0-privately. A sets a variable $pri = 1$ in this case. We let the loop halt when A finds both q_1 and q_2 are faulty (i.e., $err_1 > 1$ and $err_2 > 1$), or when A concludes that the message has been transmitted 0-privately (i.e., $pri = 1$). Based on this idea, we give a $(0, 0)$-secure message transmission protocol (PSP) that tolerates Guessing Attack to transmit a message m^A (see Fig.2).

Lemma 2 *PSP is a $(0, 0)$-secure message transmission protocol from A to B.*

Proof. First we show that PSP is 0-private. That is, the adversary X cannot learn m^A throughout the protocol. We consider the following two cases:

1. When **while loop** halts, $err_1 > 1$ and $err_2 > 1$. As we discussed before, this case means that both paths q_1 and q_2 are faulty, and X can corrupt both paths only if X chooses Z_3 to control. Thus A knows that p_3 is faulty and only transmits m_2^A via path p_1. It is straightforward that X is not able to learn m^A without knowing m_2^A.
2. When **while loop** halts, $pri = 1$. This case only happens when A receives 1 on each path q_j where $j \in \{1, 2\}$ and $err_j \leq 1$, and A will then reliably send 'OK' to B. Thus the adversary X who chooses Z_x and corrupts p_x can get only one share s_x^A, and hence cannot recover m_1^A, and simultaneously cannot learn m^A.

Thus, we showed that in both cases, m^A is transmitted 0-privately.

Next, we prove that PSP is 0-reliable. That is, B is guaranteed to recover $m^B = m^A$. It is straightforward that if X keeps passive on path p_x $(1 \leq x \leq 3)$ that it corrupts, then B can reliably recover $m_1^B = m_1^A$. Now we show that if X forwards faulty shares on p_x, then B can get $f = x$ (i.e., $p_f = p_x$). When $f = 0$ and B finds error in the received shares in Step 2, B sends 0 to A via paths q_1 and q_2. Then in Step 4, if B reliably receives m_1^A, then B can work out which path transmitted the faulty share in the previous Step 2, thus B gets $f = x$; else if B reliably receives 'OK', then it is straightforward that $f = x = 3$. Thus, B can always identify which path $p_f = p_x$ is faulty, and

A sets $err_1 := 0$, $err_2 := 0$, $pri := 0$;
B sets $f := 0$, $flag := 0$;[a]
while ($err_1 \leq 1$ **or** $err_2 \leq 1$) **and** $pri = 0$ **loop**
  A chooses an $m_1^A \in_R \mathbb{F}$ and constructs $(2,3,0)$-secret-shares (s_1^A, s_2^A, s_3^A) of m_1^A;
  Step 1 For each $1 \leq i \leq 3$, A sends s_i^A to B via path p_i;
  Step 2 B receives three shares (s_1^B, s_2^B, s_3^B);
    if $f \neq 0$ **then**
      B recovers m_1^A from shares s_i^B and s_j^B where $i, j \in \{1,2,3\} \setminus \{f\}$;
      B sends 1 to A via path q_1 and path q_2;
    else if B detects[b] 1 error in (s_1^B, s_2^B, s_3^B) **then**
      B sends 0 to A via path q_1 and path q_2, and sets $flag := 1$;
    else if B detects 0 error in (s_1^B, s_2^B, s_3^B) **then**
      B recovers m_1^A from (s_1^B, s_2^B, s_3^B), and sends 1 to A via path q_1 and path q_2;
    end if;
  Step 3 A receives $fdb_1 \in \{0,1\}$ on path q_1 and $fdb_2 \in \{0,1\}$ on path q_2;
    if $err_1 > 1$ **or** $err_2 > 1$ **then**
      A only considers fdb_h where $h \in \{1,2\}$ and $err_h \leq 1$;
      if $fdb_h = 0$ **then**
        A sets $err_h := err_h + 1$, and reliably sends m_1^A to B;
      else if $fdb_h = 1$ **then**
        A sets $pri := 1$, and reliably sends 'OK' to B;
      end if;
    else if $err_1 \leq 1$ **and** $err_2 \leq 1$ **then**
      if $fdb_1 = fdb_2 = 1$ **then**
        A sets $pri := 1$, and reliably sends 'OK' to B;
      else then
        A sets $err_h := err_h + 1$ for each $1 \leq h \leq 2$ such that $fdb_h = 0$;
        A reliably sends m_1^A to B;
      end if;
    end if;
  Step[c] 4 **if** $flag = 1$ **then**
    if B reliably receives $m_1^B := m_1^A$ **then**
      B sets $f := l$ such that s_l^B is not a correct share of m_1^B;
    else if B reliably receives 'OK' **then**
      B sets $f = 3$,[d] and recovers m_1^B from s_1^B and s_2^B;
    end if;
  end if;
end loop; - while

[a] Later in PSP, if B concludes that a path p_i $(1 \leq i \leq 3)$ is faulty, then B sets $f := i$ to mark the faulty path p_f.
[b] As we mentioned in Section 2, a $(k+1, n, 0)$-SSS can detect $n - k - 1$ errors using error-detecting code. Thus B can detect 1 error with the $(2,3,0)$-secret-shares.
[c] B does not come to Step 4 unless B sent 0 as feedback in Step 2.
[d] In this case, B knows that A did not receive 0, so B concludes that both paths $q_1 \in \bar{\mathcal{Q}}_{Z_1}$ and $q_2 \in \bar{\mathcal{Q}}_{Z_2}$ are faulty. Thus B knows that Z_3, and hence p_3, are faulty.

Fig. 2. Perfectly Secure Protocol (PSP). (*continued on next page*)

A reliably sends '$err_1 > 1$ and $err_2 > 1$' or '$pri = 1$' to B**;**
B then halts the loop and keeps the last m_1^B**;**
A sets $m_2^A := m^A - m_1^A$**;**
if $err_1 > 1$ **and** $err_2 > 1$ **then**
  A sends m_2^A to B via paths p_1**;**[e]
  B receives m_2^B on path p_1, and recovers $m^B = m_1^B + m_2^B$**;**
else if $pri = 1$ **then**
  A reliably sends m_2^A to B**;**
  B reliably receives $m_2^B = m_2^A$, and recovers $m^B = m_1^B + m_2^B$**;**
end if; - end PSP

[e] In this case, A concludes that both paths $q_1 \in \bar{\mathcal{Q}}_{Z_1}$ and $q_2 \in \bar{\mathcal{Q}}_{Z_2}$ are faulty. Thus A knows that Z_3 is faulty, so $p_1 \in \bar{\mathcal{P}}_{Z_2 \cup Z_3}$ is honest.

Fig. 2. Perfectly Secure Protocol (PSP). (*continued*)

recover $m_1^B = m_1^A$ with the shares received on the other two paths. Since it is straightforward that B can reliable receive $m_2^B = m_2^A$, B can recover $m^B = m^A$. Thus PSP is 0-reliable. □

5 $(\epsilon, 0)$-secure message transmission

In this section, we show that the necessary and sufficient conditions for achieving $(\epsilon, 0)$-secure message transmission are the same to those for achieving $(0, 0)$-secure message transmission. That is, lowering privacy level does not reduce the requirement of network connectivity. Before we prove this, we first show some results on $(k+1, n, \epsilon)$-SSS where $0 \leq \epsilon < 1$.[6] It has been discussed that a $(k+1, n, 0)$-SSS can detect $n-k-1$ errors and correct $\lfloor (n-k-1)/2 \rfloor$ errors (see [16, 17, 5]). In the following we show that a $(k+1, n, \epsilon)$-SSS can do just the same.

Lemma 3 *Let m be a secret, S be a $(k+1, n, \epsilon)$-SSS and $(v_1, ..., v_n) \in S(m)$, then any $k+1$ entries of $(v_1, ..., v_n)$ are unique to the codeword of $S(m)$.*

Proof. Assume there are some $k+1$ entries that also belong to the codeword of $S(m')$, where $m' \neq m$. Then with these $k+1$ entries, one cannot distinguish whether m or m' is shared, so m cannot be recovered with probability 1. This contradicts to property-1 of the $(k+1, n, \epsilon)$-SSS. □

Lemma 4 *Let m be a secret, S be a $(k+1, n, \epsilon)$-SSS and $(v_1, ..., v_n) \in S(m)$. For any k such entries $v_{l_1}, ..., v_{l_k}$ $(1 \leq l_1 < ... < l_k \leq n)$, there exists a secret $m' \neq m$ such that $(v'_1, ..., v'_n) \in S(m')$ and for each $1 \leq i \leq k : v'_{l_i} = v_{l_i}$.*

Proof. Assume that there are k entries $v_{l_1}, ..., v_{l_k}$ that belong to a codeword in $S(m)$, but not to any in $S(m')$, where $m' \neq m$. That is, these k entries are unique to the codeword of $S(m)$, so m can be recovered from these k entries with probability 1. This contradicts to property-2 of the $(k+1, n, \epsilon)$-SSS. □

Theorem 3 *A $(k+1, n, \epsilon)$-SSS can detect $n-k-1$ errors, but not more.*

[6] See Definition 1 in Section 2 for the definition of $(k+1, n, \epsilon)$-SSS.

Proof. Let S be a $(k+1, n, \epsilon)$-SSS and $(v_1, ..., v_n) \in S(m)$ be a codeword. First we show that if there is a tuple $T = (u_1, ..., u_n)$ such that $|\{i : u_i \neq v_i, 1 \leq i \leq n\}| = d$ and $0 < d \leq n-k-1$, then one can detect that T is not a codeword. Since $n - d \geq n - (n-k-1) = k+1$, there are at least $k+1$ entries $u_{l_1}, ..., u_{l_{k+1}}$ $(1 \leq l_1 < ... < l_{k+1} \leq n)$ such that for each $1 \leq i \leq k+1 : u_{l_i} = v_{l_i}$. Thus according to Lemma 3, $u_{l_1}, ..., u_{l_{k+1}}$ are unique to the codeword of $S(m)$. Since the d errors are not in the codeword of $S(m)$, it is easy to show that T is not a codeword.

Next we show that if $d \geq n-k$, then the tuple T can also be a codeword of a secret $m' \neq m$. Since $n - d \leq n - (n-k) = k$, there are at most k entries $u_{l_1}, ..., u_{l_k}$ $(1 \leq l_1 < ... < l_k \leq n)$ such that for each $1 \leq i \leq k : u_{l_i} = v_{l_i}$. According to Lemma 4, there exists a secret m' such that the $n-d$ entries belong to the codeword of $S(m')$, and it is possible that the d errors are also in the codeword of $S(m')$. Thus T can be codeword, and hence one cannot detect $d \geq n-k$ errors. □

Theorem 4 *A $(k+1, n, \epsilon)$-SSS can correct $\lfloor (n-k-1)/2 \rfloor$ errors, but not more.*

Proof. Let S be a $(k+1, n, \epsilon)$-SSS and $(v_1, ..., v_n) \in S(m)$ be a codeword. First we show that if there is a tuple $T = (u_1, ..., u_n)$ such that $|\{i : u_i \neq v_i, 1 \leq i \leq n\}| = c$ and $c \leq \lfloor (n-k-1)/2 \rfloor$, then one can recover the secret m from T. To correct c errors, one selects $n-c$ entries from T and put them into a new tuple T' of length $n-c$. Since $n - c \geq k+1$, T' is a corrupted codeword of a $(k+1, n-c, \epsilon)$-SSS that shares m, with at most c errors. According to Theorem 3, a $(k+1, n-c, \epsilon)$-SSS can detect

$$n - c - k - 1 \geq n - \lfloor (n-k-1)/2 \rfloor - k - 1 \geq \lfloor (n-k-1)/2 \rfloor \geq c$$

errors. With at most c errors in T', one can detect if T' is a codeword. If one finds that T' is not a codeword, it uses exhaustive search until it finds a T' that is a codeword (i.e., the c errors are not entries in T'), and finally recovers the secret m from T'.

Next we show that if $c > \lfloor (n-k-1)/2 \rfloor$, then one cannot correct c errors and recover m from T. We will construct the tuple T, in a way we explain further. Assume that $c = \lfloor (n-k-1)/2 \rfloor + 1$. Since $|\{i : u_i = v_i, 1 \leq i \leq n\}| = n - c \geq k$, according to Lemma 4, there exists a secret $m' \neq m$ such that some k error-free entries in T not only belong to a codeword $(v_1, \ldots, v_n) \in S(m)$, but also belong to a codeword of $(v'_1, \ldots, v'_n) \in S(m')$. Let us analyze the remaining $n-k$ entries of T. They consist of c errors and $c' = n-k-c$ error-free entries, i.e., c' entries identical to the corresponding ones in $(v_1, \ldots, v_n)$. We now observe that:

$$\begin{aligned} c' = n-k-c &= n-k-(\lfloor (n-k-1)/2 \rfloor + 1) \\ &\leq 2 \times \lfloor (n-k-1)/2 \rfloor + 2 - (\lfloor (n-k-1)/2 \rfloor + 1) \\ &= \lfloor (n-k-1)/2 \rfloor + 1 \\ &= c. \end{aligned}$$

We are now in a position to prove our claim. We first explain how we construct the c entries u_i in T that differ from $(v_1, \ldots, v_n)$. We let these correspond to the corresponding c entries in $(v'_1, \ldots, v'_n)$. Now since $c' \leq c$, observe that given the tuple T, one cannot distinguish whether the secret m is shared and the c entries are errors, or the secret m' is shared and the c' entries are errors. Thus cannot recover m with probability 1. □

Now we show that the conditions for achieving $(\epsilon, 0)$-secure message transmission are the same to those for achieving $(0, 0)$-security.

Theorem 5 *The CONs of Theorem 2 are also necessary and sufficient for* $(\epsilon, 0)$*-secure message transmission.*

Proof. The sufficiency of CONs is straightforward, and Patra et al.'s *Secure Protocol* in [20] is actually an $(\epsilon, 0)$-secure protocol. Now we prove the necessity of CONs, using a method similar to [5, 6].

It is straightforward that CON-1 is necessary for 0-reliable message transmission from A to B. Now we show that CON-2 is also necessary. For a contradiction, we assume that there are three sets $Z_1, Z_2, Z_3 \in \mathcal{Z}$ such that $\mathcal{Q}_{Z_1} \subsetneq \mathcal{Q}$, $\mathcal{Q}_{Z_2} = \mathcal{Q}_{Z_3} = \mathcal{Q}$ and $\mathcal{P}_{Z_1 \cup Z_2 \cup Z_3} = \mathcal{P}$. We assume an $(\epsilon, 0)$-secure message transmission protocol Π, and show how a non-threshold adversary X can defeat this protocol Π.

Let m^A be the message that A wants to send to B. X will simulate the possible behaviors of A and B by executing Π to transmit another message $\hat{m}^A \in \mathcal{M}$. The strategy of X is to flip two coins $c \in \{00, 01, 10, 11\}$:

- $c = 00$. X re-flips.
- $c = 01$. X chooses Z_1 to control, and acts passive on all paths in $\mathcal{P}_{Z_1}$ and $\mathcal{Q}_{Z_1}$.
- $c = 10$ (or $c = 11$). X chooses Z_2 (or Z_3) to control. On all paths in $\mathcal{P}_{Z_2}$ (or $\mathcal{P}_{Z_3}$), X ignores what A sends in each step of Π and simulates what A would send to B if A was sending $\hat{m}^A$. On all paths in $\mathcal{Q}_{Z_2} = \mathcal{Q}$ (or $\mathcal{Q}_{Z_3} = \mathcal{Q}$), X ignores what B sends in each step of Π and simulates what B would send to A if $c = 01$.

Note that the simulation of X on the feedback channel $\mathcal{Q}$ when $c = 10$ or $c = 11$ may not succeed, since B may send something that X fails to catch. However, there is a non-zero probability that the simulation succeeds, given X knows the protocol and can always guess. This non-zero probability can breach the 0-reliability, as we show next. It is straightforward that, when the simulation succeeds, despite what the outcome of c is, the feedbacks that A receives are the same. That is, according to the feedbacks, A will always learn that B has reliably received m^A without an error happening on the forward channel. At the end of the protocol, the view $view^B$ of B could be divided into three parts $view_{Z_1}$, $view_{Z_2}$ and $view_{Z_3}$, where $view_{Z_i}$ $(i = 1, 2, 3)$ consists of all information that paths in $\mathcal{P}_{Z_i}$ have learned (see [6]). Since the view $view^A$ of A is the same despite which set of Z_1, Z_2 or Z_3 that X chooses, and Π is ϵ-private, m^A can be recovered from any single $view_{Z_i}$ with probability at most ϵ $(\epsilon < 1)$. Thus we regard $(view_{Z_1}, view_{Z_2}, view_{Z_3})$ as shares of m^A in a $(2, 3, \epsilon)$-SSS. Next, since Π is a 0-reliable, B should be able to recover the message m^A from two of the views $(view_{Z_1}, view_{Z_2}, view_{Z_3})$ with probability 1. That is, when $c = 10$ or $c = 11$, B should be able to distinguish which view of $view_{Z_2}$ or $view_{Z_3}$ contains faulty information. To sum up, $(view_{Z_1}, view_{Z_2}, view_{Z_3})$ is a $(2, 3, \epsilon)$-SSS that can correct 1 error (either $view_{Z_2}$ or $view_{Z_3}$). According to Theorem 4, a $(2, 3, \epsilon)$-SSS can only correct $\lfloor (3 - 1 - 1)/2 \rfloor = 0$ error. We have a contradiction, which concludes the proof. □

Straightforwardly, using the result of Theorem 4 and similar proof to Theorem 5, we give the following corollary:

Corollary 1 *Let* $0 \leq \delta < \frac{1}{2}$ *and* $0 \leq \epsilon_1 < \epsilon_2 < 1$*. In any network model and any adversary model, the network connectivity required for* (ϵ_1, δ)*-secure message transmission is the same as that for* (ϵ_2, δ)*-secure message transmission.*

for $1 \leq i \leq t$ **loop**

Step 1 For each $p_j \in \bar{\mathcal{P}}_{Z_i}$, A chooses $(a^A_{i,j}, b^A_{i,j}, c^A_{i,j}) \in_R \mathbb{F}^3$ and sends the 3-tuple $(a^A_{i,j}, b^A_{i,j}, c^A_{i,j})$ to B via path p_j;

Step 2 For each $p_j \in \bar{\mathcal{P}}_{Z_i}$, B receives $(a^B_{i,j}, b^B_{i,j}, c^B_{i,j})$ on path p_j;
For each $q_j \in \bar{\mathcal{Q}}_{Z_i}$, B chooses $(d^B_{i,j}, e^B_{i,j}, f^B_{i,j}) \in_R \mathbb{F}^3$ and sends the 3-tuple $(d^B_{i,j}, e^B_{i,j}, f^B_{i,j})$ to A via path q_j;

Step 3 For each $q_j \in \bar{\mathcal{Q}}_{Z_i}$, A receives $(d^A_{i,j}, e^A_{i,j}, f^A_{i,j})$ on path q_j;
A computes $C^A := \sum_{p_j \in \bar{\mathcal{P}}_{Z_i}} a^A_{i,j} + \sum_{q_j \in \bar{\mathcal{Q}}_{Z_i}} d^A_{i,j}$, $D^A := \sum_{p_j \in \bar{\mathcal{P}}_{Z_i}} b^A_{i,j} + \sum_{q_j \in \bar{\mathcal{Q}}_{Z_i}} e^A_{i,j}$, $E^A := \sum_{p_j \in \bar{\mathcal{P}}_{Z_i}} c^A_{i,j} + \sum_{q_j \in \bar{\mathcal{Q}}_{Z_i}} f^A_{i,j}$;
A sends the 2-tuple $(M^A + E^A, auth(M^A + E^A; C^A, D^A))$ to B via all paths in $\bar{\mathcal{P}}_{Z_i}$;

Step 4 For each $p_j \in \bar{\mathcal{P}}_{Z_i}$, B receives $(g^B_{i,j}, h^B_{i,j})$ on path p_j;
if $(g^B_{i,j}, h^B_{i,j}) = (g^B_{i,k}, h^B_{i,k})$ for all $p_j, p_k \in \bar{\mathcal{P}}_{Z_i}$ **then**
B computes $C^B := \sum_{p_j \in \bar{\mathcal{P}}_{Z_i}} a^B_{i,j} + \sum_{q_j \in \bar{\mathcal{Q}}_{Z_i}} d^B_{i,j}$, $D^B := \sum_{p_j \in \bar{\mathcal{P}}_{Z_i}} b^B_{i,j} + \sum_{q_j \in \bar{\mathcal{Q}}_{Z_i}} e^B_{i,j}$, $E^B := \sum_{p_j \in \bar{\mathcal{P}}_{Z_i}} c^B_{i,j} + \sum_{q_j \in \bar{\mathcal{Q}}_{Z_i}} f^B_{i,j}$;
if $h^B_{i,j} = auth(g^B_{i,j}; C^B, D^B)$ **then**
B recovers the message $M^B := g^B_{i,j} - E^B$, and terminates the protocol;
end if;
end if;

end loop; - end APRP

Fig. 3. Almost Perfectly Reliable Protocol (APRP)

6 $(0, \delta)$-secure message transmission

In this section we discuss $(0, \delta)$-secure message transmission. Achieving probabilistic reliability has been studied extensively in the presence of a threshold adversary (see [5, 26, 24]). We use the same network model to that in [5]. Thus our result is a generalization of the results in [5], only that we consider a more general adversary structure.

Theorem 6 *Let $G(V, E)$ be a directed graph, $\mathcal{Z}$ be an adversary structure on $V \setminus \{A, B\}$, and $\mathcal{Q} \neq \emptyset$. The necessary and sufficient conditions for $(0, \delta)$-secure $(0 < \delta < \frac{1}{2})$ message transmission from A to B are:*

(i) for any set $Z_a \in \mathcal{Z}$: $\mathcal{P}_{Z_a} \subsetneq \mathcal{P}$, and

(ii) for any two sets $Z_a, Z_b \in \mathcal{Z}$: $\mathcal{P}_{Z_a \cup Z_b} \cup \mathcal{Q}_{Z_a \cup Z_b} \subsetneq \mathcal{P} \cup \mathcal{Q}$.

Proof. First we show that the conditions are necessary. It is straightforward that condition (i) must be satisfied, since it must be ensured that at least one path can transmit the correct message from A to B. To prove condition (ii) is also necessary, we assume that there are two sets $Z_1, Z_2 \in \mathcal{Z}$ such that $\mathcal{P}_{Z_1 \cup Z_2} = \mathcal{P}$ and $\mathcal{Q}_{Z_1 \cup Z_2} = \mathcal{Q}$, and there is a $(0, \delta)$-secure $(0 < \delta < \frac{1}{2})$ message transmission protocol Π. Let M^A be the message A transmits, and the adversary X chooses a faulty message $\hat{M^A}$. The strategy of X is to flip a coin and decide which set of Z_x $(x \in \{1, 2\})$ to control. In each execution step of Π, X causes each path in $\mathcal{P}_{Z_x}$ to follow the protocol as if the transmitted message is $\hat{M^A}$; if $x = 1$, then on each path in $\mathcal{Q}_{Z_1}$ (if such path exists), X simulates what B will send if B had received the faulty message $\hat{M^A}$ from paths in $\mathcal{P}_{Z_2}$ and received the actual message M^A from the other paths; else if $x = 2$, then on each path in $\bar{\mathcal{Q}}_{Z_1}$ (if such path exists), X simulates what B will send if B had received $\hat{M^A}$ from paths in $\mathcal{P}_{Z_1}$ and received M^A from the other paths.

Therefore, at the end of the protocol, A receives the same feedbacks despite whether $x = 1$ or $x = 2$. The view $view^B$ of B could divided into two parts $view_{Z_1}$ and $view_{Z_2}$, where $view_{Z_r}$ ($r \in \{1, 2\}$) consists of all information that the nodes in Z_r have learned (see similar proof in [6]). Due to the fact that the forward channel is not reliable for message transmission, B cannot distinguish whether $x = 1$ or $x = 2$, neither. Since Π is 0-private, M^A must not be recovered from any single $view_{Z_r}$. Since Π is δ-reliable, B should be able to recover the M^A from one of the two views $view_{Z_1}$ or $view_{Z_2}$ with high probability. Thus we have a contradiction.

Next we show that the conditions are sufficient. Let $\tilde{\mathcal{Z}} = \{Z_1, ..., Z_t\}$, and $M^A \in \mathcal{M}$ be the message A wants to transmit to B. We shall construct a $(0, \delta)$-secure message transmission protocol (APRP), which is similar to that in [5, Theorem 3] (see Fig.3).

Due to condition (ii), X cannot corrupt all paths in $\bar{\mathcal{P}}_{Z_i} \cup \bar{\mathcal{Q}}_{Z_i}$ for any $Z_i \in \tilde{\mathcal{Z}}$. Thus it is obvious that X cannot learn C^A, D^A and E^A in any round i of **for loop**, and hence cannot recover the message M^A. Thus APRP is 0-private.

It is straightforward that in round x, all values are transmitted via paths in $\bar{\mathcal{P}}_{Z_x} \cup \bar{\mathcal{Q}}_{Z_x}$. It is clear that in this round, B can recover $M^B = M^A$, since X who chooses Z_x can do nothing with the message transmission. The reliability is breached only if in a round i of APRP, X corrupts all paths in $\bar{\mathcal{P}}_{Z_i}$ (then X cannot corrupt all paths in $\bar{\mathcal{Q}}_{Z_i}$, due to condition (ii)), and X correctly guesses the key (C^A, D^A) with small probability. This makes APRP δ-reliable. □

Acknowledgement: The authors of this paper would like to thank the anonymous referees for their helpful comments on the earlier version of the paper.

References

1. S. Agarwal, R. Cramer, and R. de Hann. Asymptotically optimal two-round perfectly secure message transmission. In C. Dwork, editor, *CRYPTO'06 (LNCS 4117)*, pages 394–408, 2006.
2. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computing. In *Proc. ACM STOC'88*, pages 1–10. ACM Press, 1988.
3. G. Blakley. Safeguarding cryptographic keys. In *Proc. AFIPS'79 National Computer Conference*, volume 48, pages 313–317, New York, June 1979.
4. D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditional secure protocols. In *Proc. ACM STOC'88*, pages 11–19. ACM Press, 1988.
5. Y. Desmedt and Y. Wang. Perfectly secure message transmission revisited. In L.Knudsen, editor, *Proc. Eurocrypt'02 (LNCS 2332)*, pages 502–517. Springer-Verlag, April-May 2002.
6. Y. Desmedt, Y. Wang, and M. Burmester. A complete characterization of tolerable adversary structures for secure point-to-point transmissions without feedback. In X.Deng and D.Du, editors, *Proc. ISAAC'05 (LNCS 3827)*, December 2005.
7. D. Dolev. The Byzantine generals strike again. *J. of Algorithms*, 3:14–30, 1982.
8. D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *Journal of the ACM*, 40(1):17–47, January 1993.
9. M. Franklin and R. Wright. Secure communication in minimal connectivity models. *Journal of Cryptology*, 13(1):9–30, 2000.
10. E. Gilbert, F. MacWilliams, and N. Sloane. Codes which detect deception. *The BELL System Technical Journal*, 53(3):405–424, 1974.
11. V. Hadzilacos. *Issues of Fault Tolerance in Concurrent Computations*. PhD thesis, Harvard University, Cambridge, MA, 1984.
12. M. Hirt and U. Maurer. Player simulation and general adversary structures in perfect multiparty computation. *Journal of Cryptology*, 13(1):31–60, 2000.
13. M. Ito, A. Saito, and T. Nishizeki. Secret sharing schemes realizing general access structures. In *Proc. IEEE Global Telecommunications Conf., Globecom'87*, pages 99–102. IEEE Communications Soc. Press, 1987.
14. M. Kumar, P. Goundan, K. Srinathan, and C. P. Rangan. On perfectly secure communication over arbitrary networks. In *Proc. ACM PODC'02*, pages 293–202, 2002.

15. K. Kurosawa and K. Suzuki. Truly efficient 2-round perfectly secure message transmission scheme. In *Proc. Eurocrypt'08 (LNCS 4965)*, pages 324–340, 2008.
16. F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes*. North-Holland Publishing Company, 1978.
17. R. J. McEliece and D. V. Sarwate. On sharing secrets and Reed-Solomon codes. *Communications of ACM*, 24(9):583–584, September 1981.
18. A. Patra, A. Cloudhary, and C. P. Rangan. Unconditionally reliable and secure message transmission in directed networks revisited. In *SCN'08*, pages 309–326, 2008.
19. A. Patra, A. Cloudhary, and C. P. Rangan. Brief announcement: perfectly secure message transmission in directed networks re-visited. In *PODC'09*, pages 278–279, 2009.
20. A. Patra, B. Shankar, A. Choudhary, K. Srinathan, and C. P. Rangan. Perfectly secure message transmission in directed networks tolerating threshold and non threshold adversary. In *Proc. CANS*, pages 80–101, 2007.
21. T. Rabin. Robust sharing of secrets when the dealer is honest or cheating. *J. of the ACM*, 41(6):1089–1109, 1994.
22. T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proc. ACM STOC'89*, pages 73–85. ACM Press, 1989.
23. A. Shamir. How to share a secret. *Communication of ACM*, 22(11):612–613, November 1979.
24. B. Shankar, P. Gopal, K. Srinathan, and C. P. Rangan. Unconditionally reliable message transmission in directed networks. In Shang-Teng Huang, editor, *SODA*, pages 1048–1055, 2008.
25. K. Srinathan, A. Narayanan, and C. P. Rangan. Optimal perfectly secure message transmission. In *Proc. CRYPRO'04*, pages 545–561, 2004.
26. K. Srinathan and C. P. Rangan. Possibility and complexity of probabilistic reliable communications in directed networks. In *Proc. ACM PODC'06*, 2006.
27. Y. Wang and Y. Desmedt. Perfectly secure message transmission revisited. *IEEE Transaction on Information Theory*, 54(6):2582–2595, June 2008.

Appendix

A Guessing Attack to Patra et al.'s Protocols

In [20], Patra et al. proposed three protocols for secure message transmission with feedbacks: *Protocol I* and *Protocol II* were claimed to be $(0,0)$-secure against a k-active threshold adversary, and *Secure Protocol* was claimed to be $(0,0)$-secure against a general adversary structure. Here we show that neither of the three protocols enables 0-private message transmission when Guessing Attack takes place. Without loss of generality, we assume that the transmitted message $m \in_R \mathbb{F}$.

Since Protocol I and Protocol II are very similar, here we only show a Guessing Attack to Protocol I. A sketch of Protocol I (PI) is as follows.

Conditions for PI There are $u \geq 1$ directed node-disjoint path from B to A and $n = 3k - 2u + 1$ ($k \geq 2u$) directed node-disjoint paths from A to B.

Sketch of PI Let $p_1, ..., p_n$ be the directed paths from A to B, $q_1, ..., q_u$ be the directed path from B to A, and m be the message.

Phase I A selects a bivariate polynomial $Q(x,y) = \sum_{i=0}^{k} \sum_{j=0}^{k} r_{ij} x^i y^j$, where each $r_{ij} \in_R \mathbb{F}$ and $r_{00} = m$. For each $1 \leq i \leq n$, A sends the polynomial $Q(x,i)$ and the values $v_{ji} = Q(i,j)$ for $1 \leq j \leq n$ to B via path p_i.

Phase II On each p_i, B receives a polynomial $Q^B(x,i)$ and values v_{ji}^B for $1 \leq j \leq n$. For each $1 \leq i,j \leq n$, if $Q^B(j,i) \neq v_{ij}^B$, then B adds a 4-tuple $(i, j, Q^B(j,i), v_{ij}^B)$ to a list L. B finally sends the list L via all paths $q_1, ..., q_u$.

Phase III A receives $l \leq u$ distinct lists $L_1, ..., L_l$ on the feedback channels. For each list L_t ($1 \leq t \leq l$), A creates a list $L_{t_{fault}}$. For each 4-tuple $(i^A, j^A, Q^A(j^A, i^A), v_{i^A j^A}^A)$ in

The adversary X chooses to control paths $p_1, ..., p_{k-u}$ and all paths $q_1, ..., q_u$. Thus X is able to get polynomials $(Q(x,1), ..., Q(x,k-u))$ and matrix $V = (v_{ji})_{n\times(k-u)}$ in Phase I.
The strategy of X is to act passive in Phase I of PI. In Phase II, X guesses the $3u+1$ polynomials transmitted on paths $p_{k-u+1}, ..., p_{k+2u+1}$.[a] For each $1 \le t \le u$, X sets $base_t = \{p_{k-u+t}, p_{k+t}\}$, and also X sets a set $test = \{p_{k+u+1}, ..., p_{k+2u+1}\}$. Thus the $3u+1$ paths $p_{k-u+1}, ..., p_{k+2u+1}$ are divided into u sets of size 2 and a set $test$ such that $|test| = u+1$.
For each $1 \le t \le u$, let $\{p_a, p_b\} = base_t$, X chooses at random two polynomials $Q^X(x,a)$ and $Q^X(x,b)$ and guarantees $Q^X(l,a) = v_{al}$ and $Q^X(l,b) = v_{bl}$ for each $1 \le l \le k-u$ (X knows v_{al} and v_{bl} on path p_l $(1 \le l \le k-u)$). For each $k+u+1 \le j \le k+2u+1$ (thus $p_j \in test$), X chooses two values $v^X_{aj}, v^X_{bj} \in_R \mathbb{F}$, and adds two 4-tuples $(a, j, Q^X(j,a), v^X_{aj})$ and $(b, j, Q^X(j,b), v^X_{bj})$ to a list L^X_t). Finally, X transmits L^X_t ($|L^X_t| = 2u+2$) to A via path q_i.
In Phase III, corresponding to each L^X_t $(1 \le t \le u)$, A broadcasts a pair $(L_t, L_{t_{fault}})$. If there exists a $p_a \in base_t$ and $p_a \notin L_{t_{fault}}$, then X knows that all the $u+1$ values of $Q^X(j,a)$ $(k+u+1 \le j \le k+2u+1)$ are correct, thus $Q^X(x,a) = Q(x,a)$; if there exists a $p_j \in test$ and $p_j \notin L_{t_{fault}}$, then X knows that both the two values v^X_{aj} and v^X_{bj} are correct, where $(\{p_a, p_b\} = base_t)$, thus X gets $Q(j,a)$ and $Q(j,b)$ that are useful for reconstructing $Q(j,y)$.
After PI, if X gets $u+1$ polynomials of $Q(x,i)$ $(k-u+1 \le i \le k+u)$ or X gets $u+1$ polynomials $Q(j,y)$ $(k+u+1 \le j \le k+2u+1)$, then X reconstructs $Q(x,y)$ and recovers m; otherwise, X uses a random guess to get a m'.

[a] Since $k \ge 2u$, we have $n = 3k-2u+1 \ge k+4u-2u+1 = k+2u+1$, thus such paths exist.

Fig. 4. Guessing Attack to PI.

the list L_t, A checks whether $Q^A(j^A, i^A) = Q(j^A, i^A)$ and whether $v^A_{i^A j^A} = v_{i^A j^A}$. If $Q^A(j^A, i^A) \ne Q(j^A, i^A)$, then A adds p_{i^A} to $L_{t_{fault}}$; if $v^A_{i^A j^A} \ne v_{i^A j^A}$, then A adds p_{j^A} to $L_{t_{fault}}$.
Finally, for each $1 \le t \le l$, A broadcasts the pairs $(L_t, L_{t_{fault}})$.

...

We show that PI is not 0-private (i.e., contradict to [20, Theorem 2]) by the Guessing Attack in Fig.4.

We observe that by performing the Guessing Attack shown in Fig.4, X can recover m by the following two ways:

W-1 X reconstructs the coefficient of x^0 in the polynomial $Q(x,y)$. That is, X needs another $u+1$ polynomials from $(Q(x,k-u+1), ..., Q(x,n))$. For each polynomial $Q(x,i)$ $(k-u+1 \le i \le n)$, since X has already obtained the values $(Q(1,i), ..., Q(k-u,i))$ (i.e., $(v_{i1}, ..., v_{i(k-u)})$), it needs only $u+1$ values of $Q(j,i)$ $(k-u+1 \le j \le n)$ to reconstruct $Q(x,i)$;

W-2 X reconstructs the coefficient of y^0 in the polynomial $Q(x,y)$. Since on each path p_j $(1 \le j \le k-u)$, X gets $Q(j,1), ..., Q(j,n)$, we know that X can reconstruct the polynomial $Q(j,y)$. Thus X has obtained $k-u$ polynomials $(Q(1,y), ..., Q(k-u,y))$ and needs another $u+1$ polynomials from $(Q(k-u+1,y), ..., Q(n,y))$. For each polynomial $Q(j,y)$ $(k-u+1 \le j \le n)$, since X has already obtained the values $(Q(j,1), ..., Q(j,k-u))$ from the polynomials $(Q(x,1), ..., Q(x,k-u))$, it needs only $u+1$ values of $Q(j,i)$ $(k-u+1 \le i \le n)$ to reconstruct $Q(j,y)$.

Therefore, X gets a $Q(x,i)$ $(k-u+1 \le i \le k+u)$ only if all the $u+1$ values of $Q^X(j,i)$ $(k+u+1 \le j \le k+2u+1)$ are correct, and since these values are independent, we have the probability P_1 with

which X gets the polynomial $Q(x,i)$ is

$$P_1 = \left(\frac{1}{|\mathbb{F}|}\right)^{u+1}.$$

We assume that X reconstructs $Q(x,y)$ using W-1. That is, out of the $2u$ polynomials $Q^X(x,i)$ $(k-u+1 \leq i \leq k+u)$ that X guesses, at least $u+1$ are correct. The probability P_2 to correctly reconstruct $Q(x,y)$ is

$$\begin{aligned} P_2 = & \tbinom{2u}{u+1} \times P_1^{u+1} \times (1-P_1)^{u-1} \\ & + \tbinom{2u}{u+2} \times P_1^{u+2} \times (1-P_1)^{u-2} + \\ & \vdots \\ & + \tbinom{2u}{2u} \times P_1^{2u}. \end{aligned}$$

For each $p_j \in test$, X needs to guess two random values for each path q_t $(1 \leq t \leq u)$, and if $p_j \notin L_{t_{fault}}$, then both the guess must be correct. Thus, in total u pairs of values will be guessed corresponding to the polynomial $Q(j,y)$ $(k+u+1 \leq j \leq k+2u+1)$, and at least $u+1$ correct guesses of v_{yj}^X-s are required to reconstruct $Q(j,y)$. Thus the probability P_3 with which X gets a $Q(j,y)$ is

$$\begin{aligned} P_3 = & \tbinom{u}{\lfloor \frac{(u+1)+1}{2} \rfloor} \times \left(\tfrac{1}{|\mathbb{F}|^2}\right)^{\lfloor \frac{(u+1)+1}{2} \rfloor} \times \left(1 - \tfrac{1}{|\mathbb{F}|^2}\right)^{\left(u - \lfloor \frac{(u+1)+1}{2} \rfloor\right)} \\ & + \tbinom{u}{\lfloor \frac{(u+1)+1}{2} \rfloor + 1} \times \left(\tfrac{1}{|\mathbb{F}|^2}\right)^{\left(\lfloor \frac{(u+1)+1}{2} \rfloor + 1\right)} \times \left(1 - \tfrac{1}{|\mathbb{F}|^2}\right)^{\left(u - \lfloor \frac{(u+1)+1}{2} \rfloor - 1\right)} + \\ & \vdots \\ & + \tbinom{u}{u} \times \left(\tfrac{1}{|\mathbb{F}|^2}\right)^{u}. \end{aligned}$$

We assume that X reconstructs $Q(x,y)$ using W-2. That is, all the $u+1$ polynomials $Q^X(j,y)$ $(k+u+1 \leq j \leq k+2u+1)$ that X guesses must be correct. Thus the probability P_4 that X reconstructs $Q(x,y)$ with is

$$P_4 = P_3^{u+1}.$$

Therefore, by using W-1 or[7] W-2, the probability T that X successfully get m with is

$$T = P_2 + P_4 - P_2 P_4.$$

If the guess fails, X will use a random guess to get an m' with probability $\frac{1}{|\mathbb{F}|}$ such that $m' = m$ (without loss of generality, we assume $m \in_R \mathbb{F}$). Thus the total probability G that X learns m using the Guessing Attack is

$$G = T + (1-T) \times \frac{1}{|\mathbb{F}|}.$$

Thus we proved that PI is not 0-private.

Now, we prove that Secure Protocol (SP), which is a three phase protocol tolerating a subset $\mathcal{B}$ of an adversary structure $\mathcal{Z}$ where $|\tilde{\mathcal{B}}| = 3$, is not 0-private. To show our Guessing Attack, we first sketch SP in the following.

[7] We observe that W-1 and W-2 can be used together. E.g., the reconstruction of one $Q(x,i)$ gives better probability to reconstruct a $Q(j,y)$, because X can get an extra $Q(j,i)$ from $Q(x,i)$, and hence needs only u values of $Q(j,y)$ (instead of $u+1$ values) to reconstruct the polynomial $Q(j,y)$. Since the goal here is only to show the Guessing Attack to PI, we do not calculate this probability here.

The adversary X chooses Z_3 to control; that is, X corrupts both q_1 and q_2. In Phase I of SP, X can only get $Q(x,3)$, with which X knows $Q(1,3)$ and $Q(2,3)$, thus it only needs the value of $Q(1,2)$ to recover m. In each phase of SP, X acts passive on paths in $\mathcal{P}_{Z_3}$. Thus B does not use the feedback channel throughout the protocol. In Phase II of SP, X chooses four distinct random numbers $v_1^X, v_2^X, v_3^X, v_4^X \in_R \mathbb{F}$, and transmits two 4-tuples $(1,2,v_1^X,v_2^X)$ and $(1,2,v_3^X,v_4^X)$ to A. Then in Phase III, if corresponding to a value v_i^X $(1 \le i \le 4)$, no appended error message "Path γ is faulty" (γ is either p_1 or p_2) is broadcast by A, then X knows that v_i^X is correct (i.e., $= Q(1,2)$), and hence recovers m; otherwise, X uses a random guess over $\mathbb{F} \setminus \{v_1^X, v_2^X, v_3^X, v_4^X\}$ to get an m'.

Fig. 5. Guessing Attack to SP.

Conditions for SP Let $\tilde{\mathcal{B}} = \{Z_1, Z_2, Z_3\}$. (1) there is a PRMT (perfectly reliable message transmission) protocol from A to B, and (2) if $\mathcal{P}_{Z_1 \cup Z_2 \cup Z_3} = \mathcal{P}$, then there exist two paths $q_\alpha \in \bar{\mathcal{Q}}_{Z_\alpha}, q_\beta \in \bar{\mathcal{Q}}_{Z_\beta}$ $(\alpha, \beta \in \{1,2,3\})$.

Sketch of SP Due to the existence of PRMT, there exist three paths $p_1 \in \bar{\mathcal{P}}_{Z_2 \cup Z_3}$, $p_2 \in \bar{\mathcal{P}}_{Z_1 \cup Z_3}$, and $p_3 \in \bar{\mathcal{P}}_{Z_1 \cup Z_2}$ (see [6]). Let m be the message that A transmits to B.

Phase I A chooses a bivariate polynomial $Q(x,y) = \sum_{i=0}^{1} \sum_{j=0}^{1} r_{i,j} x^i y^j$ uniformly at random such that $Q(0,0) = m$. $Q(x,y)$ is symmetric; i.e., $Q(i,j) = Q(j,i)$. A sends the polynomial $Q(x,i)$ to B via path p_i, $1 \le i \le 3$.

Phase II B receives the polynomial $Q_i^B(x) = Q^B(x,i)$ on path p_i, $1 \le i \le 3$. Out of the three $Q_i^B(x)$-s, at most one is corrupted. B then performs tests to determine which path p_i is faulty.[8] According to the outcome of the tests:

- if B concludes that all p_i-s $(1 \le i \le 3)$ are honest, then B recovers m and terminates the protocol;
- if B finds which p_i $(1 \le i \le 3)$ is faulty, then B recovers m and terminates the protocol;
- if B finds one of the two paths p_i and p_j $(1 \le i, j \le 3$ and $i \ne j)$ is faulty but cannot distinguish which one, then B sends a 4-tuple $(i, j, Q_i^B(j), Q_j^B(i))$ to A via paths q_α and q_β.

Phase III A receives two 4-tuples: $(i_\alpha, j_\alpha, v_{i_\alpha}, v_{j_\alpha})$ on path q_α and $(i_\beta, j_\beta, v_{i_\beta}, v_{j_\beta})$ on path q_β.

- Corresponding to $(i_\alpha, j_\alpha, v_{i_\alpha}, v_{j_\alpha})$, A checks whether $v_{i_\alpha} = Q(j_\alpha, i_\alpha)$ and whether $v_{j_\alpha} = Q(i_\alpha, j_\alpha)$. Depending on the outcome, A concludes which path p_{i_α} or p_{j_α} is faulty, and appends an error message "Path γ is faulty" (γ is either p_{i_α} or p_{j_α}) to $(i_\alpha, j_\alpha, v_{i_\alpha}, v_{j_\alpha})$.
- A performs similar computation to the other 4-tuple $(i_\beta, j_\beta, v_{i_\beta}, v_{j_\beta})$.
- A broadcasts the two 4-tuples along with the appended error messages.

...

Next we show that the adversary X can learn the message m by performing Guessing Attack (contradict to [20, Lemma 12]).

We assume there exist a path $q_1 \in \bar{\mathcal{Q}}_{Z_1}$ and a path $q_2 \in \bar{\mathcal{Q}}_{Z_2}$, and $q_1, q_2 \in \mathcal{Q}_{Z_3}$. We show that by performing the Guessing Attack in Fig.5, X can learn m with probability better than $\frac{1}{|\mathbb{F}|}$.

In this Guessing Attack, the guess is successful if there is a $v_i^X = Q(1,2) = Q(2,1)$ $(1 \le i \le 4)$, so A will broadcast the error message that indicates the value of $Q(1,2)$ to X. Thus the probability T that the guess is successful is

$$T = 4 \times \frac{1}{|\mathbb{F}|} = \frac{4}{|\mathbb{F}|}.$$

[8] The details of the tests are not important here. For more details see [20, Secure Protocol].

If the guess fails, then X knows that neither of the four random numbers it chose is correct, so it will use a random guess over $\mathbb{F} \setminus \{v_1^X, v_2^X, v_3^X, v_4^X\}$ and with probability $\frac{1}{|\mathbb{F}|-4}$, it will learn the message m. Thus the total probability G that X learns m using Guessing Attack is

$$G = T + (1-T) \times \frac{1}{|\mathbb{F}| - 4} = \frac{4}{|\mathbb{F}|} + \left(1 - \frac{4}{|\mathbb{F}|}\right) \times \frac{1}{|\mathbb{F}| - 4} = \frac{5}{|\mathbb{F}|}.$$

It is straightforward that the probability that X learns m is much higher than expected (i.e., $\frac{1}{|\mathbb{F}|}$), thus SP is not 0-private.