Perfectly Secure Message Transmission Against Independent Rational Adversaries

Kenji Yasunaga[1](✉) and Takeshi Koshiba[2]

[1] Graduate School of Information Science and Technology, Osaka University, Osaka, Japan
yasunaga@ist.osaka-u.ac.jp

[2] Faculty of Education and Integrated Arts and Sciences, Waseda University, Tokyo, Japan
tkoshiba@waseda.jp

Abstract. Secure Message Transmission (SMT) is a two-party protocol by which the sender can privately transmit a message to the receiver through multiple channels. An adversary can corrupt a subset of channels and makes eavesdropping and tampering over the corrupted channels. Fujita et al. (GameSec 2018) introduced a game-theoretic security notion of SMT, and showed protocols that are secure even if an adversary corrupts all but one of the channels, which is impossible in the standard cryptographic setting. In this work, we study a game-theoretic setting in which all the channels are corrupted by two or more independent adversaries. Specifically, we assume that there are several adversaries who exclusively corrupt subsets of the channels, and prefer to violate the security of SMT with being undetected. Additionally, we assume that each adversary prefers other adversaries' tampering to be detected. We show that secure SMT protocols can be constructed even if all the channels are corrupted by such rational adversaries. We also study the situation in which both malicious and rational adversaries exist.

Keywords: Cryptography · Secure message transmission · Game theory · Rational adversary

1 Introduction

Cryptography in the traditional sense provides the confidentiality of messages between two parties, Alice and Bob. Symmetric-key cryptography requires to share the key before communication, and the key agreement is still a problem to be resolved. Asymmetric-key cryptography (a.k.a. public-key cryptography) is free from the key agreement problem but must rely on some unproven computational hardness of mathematical problems. In the standard setting, we implicitly assume that there is a single channel between Alice and Bob. Since computer networks nowadays are like a web, we may assume that several channels are

T. Alpcan et al. (Eds.): GameSec 2019, LNCS 11836, pp. 563–582, 2019.
https://doi.org/10.1007/978-3-030-32430-8_33

available for communication between Alice and Bob. Secure message transmission (SMT) is a scheme for the communication between Alice and Bob in the environment in which several channels are available.

SMT is a two-party cryptographic protocol with n channels by which a sender Alice securely and reliably sends messages to a receiver Bob. SMT also assumes the existence of the adversary who can corrupt t channels out of the n channels. The adversary can eavesdrop messages from the corrupted channels and alter them. We consider privacy and reliability as properties of SMT against the adversaries. The privacy means that the adversary can obtain no information on the messages Alice sends to Bob. The reliability means that a message Bob receives coincides with the message Alice sends. An SMT protocol is said to be *perfect* if the protocol satisfies both properties in the perfect sense. An SMT protocol is said to be *almost-reliable* if the protocol satisfies the perfect privacy and allows transmission errors of small probability.

The notion of SMT was originally proposed by Dolev, Dwork, Waarts, and Yung [9]. They showed that any 1-round (i.e., non-interactive) perfect SMT must satisfy that $t < n/3$, and any perfect SMT of at least two rounds must satisfy that $t < n/2$. Since then, the efficiency of perfect SMT has been improved in the literature [3,25,28,32]. The most efficient 2-round perfect SMT was given by Spini and Zémor [31]. In the case of almost-reliable SMT, the situation is different from the case of perfect SMT. Franklin and Wright [10] showed an almost-reliable SMT against $t < n$ corruptions by using a public channel in addition to the usual channels. Later, Garay and Ostrovsky [15] and Shi et al. [30] gave the most round-efficient almost-reliable SMT protocols using public channels.

In the standard cryptographic setting, adversaries are assumed to be semi-honest or malicious. Semi-honest adversaries follow the protocol but try to extract secret information during the protocol execution. Malicious ones deviate from the protocol either to obtain secret information or to obstruct the protocol execution. Especially, malicious adversaries would do anything regardless of their risks. However, some adversaries realistically take their risks into account and rationally behave forward the other participants in the protocol. To incorporate the notion of "rationality" into cryptography, we employ game-theoretic ideas. Halpern and Teague [20] firstly investigated the power and the limitation of rational participants in secret sharing. Since then, rational secret sharing has been investigated in the literature [1,6,11,24]. Besides secret sharing, rational settings have been employed in other cryptographic protocols such as leader election [2,16], agreement protocols [18,21], public-key cryptography [34,35], two-party computation [5,17], delegated computation [7,19,23], and protocol design [13,14]. In particular, we can overcome the "impossibility barrier" in some cases [4,12,18] by considering that the adversaries rationally behave.

Fujita, Yasunaga, and Koshiba [12] studied a game-theoretic security model for SMT. They introduced rational "timid" adversaries who prefer to violate the security requirement of SMT but do not prefer the tampering actions to be detected. They showed that even if the adversary corrupts all but one of the channels, it is possible to construct perfect SMT protocols against rational timid adversaries. In the standard cryptographic setting, perfect SMT can be constructed only when the adversary corrupts a minority of the channels. This demonstrates a way of circumventing the impossibility results of cryptographic protocols based on a game-theoretic approach. In this paper, we further investigate the game-theoretic security of SMT.

In [12], the simplest game-theoretic setting (i.e., 1-player game) was employed. In the 1-player game, the player's behavior is determined by the strategy of the largest expected utility. In this paper, we consider the case of games for two or more players (i.e., adversaries). We study a game-theoretic setting in which all the channels may be corrupted by two or more *independent* rational timid adversaries. More specifically, we assume that there are more than one adversaries who exclusively corrupt subsets of the channels, and prefer to violate the security of SMT with being undetected. Additionally, we assume that each adversary prefers other adversaries' tampering to be detected. Note that if a single adversary corrupts all the channels, we cannot hope for the security of SMT. We show that secure SMT protocols can be constructed even if all the channels are corrupted by such independent rational adversaries. One protocol uses a public channel, and the others do not.

- We show that Shi et al.'s almost-reliable SMT protocol (after a minor adaptation) in [30], which uses a public channel, works as a perfect SMT against multiple independent rational adversaries. We assume that there are $\lambda \geq 2$ adversaries, and adversaries $i \in \{1, \dots, \lambda\}$ exclusively corrupt $t_i \geq 1$ channels such that $t_1 + \cdots + t_\lambda \leq n$.
  Since we employ a Nash equilibrium as a solution concept, the result is not surprising. Nash equilibrium requires that no deviation increases the utility, assuming that the other adversaries follow the prescribed strategy. Since the security against a single adversary corrupting $n-1$ channels is provided in [12], a similar argument can be applied in our setting, though slightly different utility functions should be considered.
- To construct perfect SMT protocols without public channel, we employ the idea of *cheater-identifiable* secret sharing (CISS), where every player who submits a forged share in the reconstruction phase can be identified. Intuitively, in the setting of rational SMT, timid adversaries will not tamper with shares because the tampering action will be detected with high probability, but the message can be recovered by using other shares. We construct a non-interactive SMT protocol based on the idea of CISS due to Hayashi and Koshiba [22]. Technically, our construction employs pairwise independent (a.k.a. strongly universal) hash functions as hash functions. Since the security requirements of CISS are not sufficient for proving the security of rational SMT, we provide the security analysis of our protocol, not for general CISS-based SMT protocols.
- The limitation of CISS is that the number of forged shares should be a minority. Namely, the above construction only works for adversaries who corrupt at most $\lfloor (n-1)/2 \rfloor$ channels. We show that a slight modification of the CISS-

based protocol gives a perfect SMT protocol against *strictly* timid adversaries even if one of them may corrupt a majority of the channels. Adversaries are said to be strictly timid if they prefer being tampering undetected to violating the reliability. A similar idea was used in the previous work of [12], where *robust* secret sharing is employed for the protocol against a strictly timid adversary. Since we consider independent adversaries who prefer other adversaries to be detected, CISS is suitable in this setting.

- Finally, we consider the setting in which a malicious adversary exists as well as rational adversaries. Namely, there are several adversaries, all but one behave rationally, but one behaves maliciously. We believe this setting is preferable because the assumption that all of the adversaries are rational may not be realistic. Mixing of malicious and rational adversaries was studied in the context of rational secret sharing [1,24]. We show that a modification of the CISS-based protocol achieves a non-interactive perfect SMT protocol against such adversaries. The protocol is secure as long as a malicious adversary corrupts $t^* \leq \lfloor (n-1)/3 \rfloor$ channels, and each rational adversary corrupts at most $\min\{\lfloor (n-1)/2 \rfloor - t^*, \lfloor (n-1)/3 \rfloor\}$ channels.

We clarify the differences from the previous work of [12]. In [12], there is only one adversary who corrupts at most $n-1$ channels. This setting can be seen as one in which there are two independent adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$. While $\mathcal{A}_1$ tries to violate the security of the SMT protocol by corrupting at most $t \leq n-1$ channels, the other adversary $\mathcal{A}_2$, who corrupt $n-t \geq 1$ channels, does nothing for the protocol. Thus, the setting of [12] can be seen as a weaker setting of independent adversaries. In other words, this work provides stronger results for the problem of SMT protocols against rational adversaries. The mixed setting of malicious and rational adversaries in this work is closest to the traditional cryptographic setting of SMT. Even in this setting, we present a non-interactive protocol against adversaries corrupting in total $t < n/2$ channels, for which cryptographic SMT requires interaction or a weaker bound $t < n/3$.

2 Secure Message Transmission

A sender $\mathcal{S}$ and a receiver $\mathcal{R}$ are connected by n channels, and in addition, they may use an authentic and reliable *public channel.* Messages sent over the public channel are publicly accessible and correctly delivered to the receiver. We assume that SMT protocols proceed in *rounds.* In each round, one party can synchronously send messages over the n channels and the public channel. The messages will be delivered before the next round starts.

The adversary $\mathcal{A}$ can corrupt at most t channels. Such an adversary is referred to as *t-adversary.* Messages sent over corrupted channels can be eavesdropped and tampered by the adversary. We assume that the adversary cannot delay messages over the corrupted channels. Namely, the tampered messages will be transmitted to the receiver in the same round. We also assume that $\mathcal{A}$ is computationally *unbounded.*

Let $\mathcal{M}$ be the message space. In SMT, the sender tries to send a message in $\mathcal{M}$ to the receiver by using n channels and the public channel, and the receiver outputs some message after the protocol execution. For an SMT protocol Π, let M_S denote the random variable of the message sent by $\mathcal{S}$ and M_R the message output by $\mathcal{R}$ in Π. An execution of Π can be completely characterized by the random coins of all the parties, namely, $\mathcal{S}$, $\mathcal{M}$, and $\mathcal{A}$, and the message M_S sent by $\mathcal{S}$. Let $V_A(m, r_A)$ denote the *view* of $\mathcal{A}$ when the protocol is executed with $M_S = m$ and the random coins r_A of $\mathcal{A}$. Specifically, $V_A(m, r_A)$ consists of the messages sent over the corrupted channels and the public channel when the protocol is run with $M_S = m$ and $\mathcal{A}$'s random coins r_A.

We formally define the properties of SMT protocols.

Definition 1. *A protocol between $\mathcal{S}$ and $\mathcal{R}$ is* (ε, δ)-Secure Message Transmission (SMT) against t-adversary *if the following three conditions are satisfied against any t-adversary $\mathcal{A}$.*

- Correctness*: For any $m \in \mathcal{M}$, if $M_S = m$ and $\mathcal{A}$ does not corrupt any channels, then* $\Pr[M_R = m] = 1$*,*
- Privacy*: For any $m_0, m_1 \in \mathcal{M}$ and $r_A \in \{0,1\}^*$, it holds that*

$$\mathrm{SD}(V_A(m_0, r_A), V_A(m_1, r_A)) \leq \varepsilon,$$

where $\mathrm{SD}(X, Y)$ *denotes the statistical distance between two random variables X and Y over a set Ω, which is defined by*

$$\mathrm{SD}(X, Y) = \frac{1}{2} \sum_{u \in \Omega} \left| \Pr[X = u] - \Pr[Y = u] \right|,$$

and
- Reliability*: For any message $m \in \mathcal{M}$, when $M_S = m$,*

$$\Pr[M_R \neq m] \leq \delta,$$

where the probability is taken over the random coins of $\mathcal{S}$, $\mathcal{R}$, and $\mathcal{A}$.

If a protocol achieves $(0, 0)$-SMT, the protocol is called *perfect* SMT, and if a protocol achieves $(0, \delta)$-SMT, which admits transmission failures of small probability δ, the protocol is called *almost-reliable* SMT.

For perfect SMT, Dolev *et al.* [9] showed the below.

Theorem 1 ([9]). *Perfect SMT protocols against t-adversary are achievable if and only if $t < n/2$.*

3 SMT Against Independent Rational Adversaries

We define our security model of SMT in the presence of independent rational adversaries. Rationality of the adversary is characterized by a *utility function* which represents the preference of the adversary over possible outcomes of the protocol execution.

We can consider various preferences of adversaries regarding the SMT protocol execution. The adversaries may prefer to violate the security of SMT protocols without the detection of tampering actions. In addition, they may prefer other adversaries to be detected by tampering actions. Here, we consider the adversaries who prefer (1) to violate the privacy, (2) to violate the reliability, (3) their tampering actions to be undetected, and (4) other adversaries' actions to be detected.

To define the utility function, we specify the SMT game as follows. We assume that there are λ adversaries $1, 2, \ldots, \lambda$ for $\lambda \geq 2$. Each adversary does not cooperate with other adversaries. We assume that adversary $j \in \{1, \ldots, \lambda\}$ exclusively corrupt at most t_j channels out of the n channels for $t_j \geq 1$, and that $\sum_{j=1}^{\lambda} t_j \leq n$.

The SMT Game. First, set parameters $\mathsf{suc} = 0$ and $\mathsf{guess}_j = \mathsf{detect}_j = 0$ for every $j \in \{1, \ldots, \lambda\}$. Given an SMT protocol Π with the message space $\mathcal{M}$, choose $m \in \mathcal{M}$ uniformly at random, and run the protocol Π in which the message to be sent is $M_S = m$. In the protocol execution, adversaries j can exclusively corrupt t_j channels, and tamper with any messages sent over the corrupted channels. The sender or the receiver may send a special message "DETECT at i" for $i \in \{1, \ldots, n\}$, meaning that some tampering action was detected at channel i. Then, if adversary j corrupts channel i, set $\mathsf{detect}_j = 1$. After running the protocol, the receiver outputs M_R, and each adversary j outputs M_j for $j \in \{1, \ldots, \lambda\}$. If $M_R = M_S$, set $\mathsf{suc} = 1$. For $j \in \{1, \ldots, \lambda\}$, if $M_j = M_S$, set $\mathsf{guess}_j = 1$. The outcome of the game is $\big(\mathsf{suc}, \{\mathsf{guess}_{j'}, \mathsf{detect}_{j'}\}_{j' \in \{1, \ldots, \lambda\}}\big)$.

The utility of the adversary is defined as the expected utility in the SMT game.

Definition 2 (Utility). *The utility $U_j(\mathcal{A}_1, \ldots, \mathcal{A}_\lambda, U)$ of adversary j when strategy $(\mathcal{A}_1, \ldots, \mathcal{A}_\lambda)$ and utility function U are employed is the expected value $E[U(j, \mathsf{out})]$, where U is a function that maps index j and the outcome $\mathsf{out} = \big(\mathsf{suc}, \{\mathsf{guess}_{j'}, \mathsf{detect}_{j'}\}_{j' \in \{1, \ldots, \lambda\}}\big)$ of the SMT game to real values, and the probability is taken over the random coins of the sender, the receiver, and the adversaries, and a random choice of message M_S.*

Each adversary $j \in \{1, \ldots, \lambda\}$ tries to maximize utility U_j by choosing a strategy A_j. Since the utility depends on other adversaries' strategies, we use game-theoretic notions in the security definition. We define the security of rational secure message transmission (RSMT). For strategies $\mathcal{B}_1, \ldots, \mathcal{B}_\lambda, \mathcal{A}_j$, we denote by $(\mathcal{A}_j, \mathcal{B}_{-j})$ the strategy profile $(\mathcal{B}_1, \ldots, \mathcal{B}_{j-1}, \mathcal{A}_j, \mathcal{B}_{j+1}, \ldots, \mathcal{B}_\lambda)$.

Definition 3 (Security of RSMT). *An SMT protocol Π is* perfectly secure against rational $(t_1, \ldots, t_\lambda)$-adversaries with utility function U *if there are t_j-adversary $\mathcal{B}_j$ for $j \in \{1, \ldots, \lambda\}$ such that for any t_j-adversary $\mathcal{A}_j$ for $j \in \{1, \ldots, \lambda\}$,*

1. Perfect security*: Π is* $(0,0)$*-SMT against* $(\mathcal{B}_1,\ldots,\mathcal{B}_\lambda)$*, and*
2. Nash equilibrium*:* $U_j(\mathcal{A}_j,\mathcal{B}_{-j},U) \leq U_j(\mathcal{B}_j,\mathcal{B}_{-j},U)$ *for every* $j \in \{1,\ldots,\lambda\}$ *in the SMT game.*

The perfect security guarantees that the strategy profile $(\mathcal{B}_1,\ldots,\mathcal{B}_\lambda)$ is *harmless*. The Nash equilibrium guarantees that no adversary j can gain more utility by changing the strategy from $\mathcal{B}_j$ to $\mathcal{A}_j$. Thus, the above security implies that each adversary j has no incentive to deviate from the harmless strategy $\mathcal{B}_j$.

In the security proof of our protocol, we will consider the strategy profile $(\mathcal{B}_1,\ldots,\mathcal{B}_\lambda)$ in which each adversary j does not corrupt any channels, and outputs M_j by choosing a message uniformly at random from $\mathcal{M}$. For such $(\mathcal{B}_1,\ldots,\mathcal{B}_\lambda)$, the perfect privacy and reliability immediately follow if Π satisfies the correctness.

Timid Adversaries

We construct secure protocols against independent *timid* adversaries, who do not prefer the tampering actions to be detected, and prefer to violate the reliability.

Regarding the utility function, let $U^{\mathsf{ind}}_{\mathsf{timid}}$ be the set of utility functions that satisfy the following conditions:

1. $U(j,\mathsf{out}) > U(j,\mathsf{out}')$ if $\mathsf{suc} < \mathsf{suc}'$, $\mathsf{guess}_j = \mathsf{guess}'_j$, and $\mathsf{detect}_j = \mathsf{detect}'_j$,
2. $U(j,\mathsf{out}) > U(j,\mathsf{out}')$ if $\mathsf{suc} = \mathsf{suc}'$, $\mathsf{guess}_j = \mathsf{guess}'_j$, $\mathsf{detect}_j < \mathsf{detect}'_j$, and $\mathsf{detect}_k = \mathsf{detect}'_k$ for every $k \in \{1,\ldots,\lambda\} \setminus \{j\}$, and
3. $U(j,\mathsf{out}) > U(j,\mathsf{out}')$ if $\mathsf{suc} = \mathsf{suc}'$, $\mathsf{guess}_j = \mathsf{guess}'_j$, $\mathsf{detect}_k > \mathsf{detect}'_k$ for some $k \neq j$, and $\mathsf{detect}_{j'} = \mathsf{detect}'_{j'}$ for every $j' \in \{1,\ldots,\lambda\} \setminus \{k\}$,

where $\mathsf{out} = (\mathsf{suc}, \{\mathsf{guess}_j, \mathsf{detect}_j\}_{j\in\{1,\ldots,\lambda\}})$ and $\mathsf{out}' = (\mathsf{suc}', \{\mathsf{guess}'_j, \mathsf{detect}'_j\}_{j\in\{1,\ldots,\lambda\}})$ are the outcomes of the SMT game.

In addition, timid adversaries may have the following property:

4. $U(j,\mathsf{out}) > U(j,\mathsf{out}')$ if $\mathsf{suc} > \mathsf{suc}'$, $\mathsf{guess}_j = \mathsf{guess}'_j$, $\mathsf{detect}_j < \mathsf{detect}'_j$, and $\mathsf{detect}_k = \mathsf{detect}'_k$ for every $k \in \{1,\ldots,\lambda\} \setminus \{j\}$.

Let $U^{\mathsf{ind}}_{\mathsf{st\text{-}timid}}$ be the set of utility functions satisfying the above four conditions. An adversary is said to be *timid* if his utility function is in $U^{\mathsf{ind}}_{\mathsf{timid}}$, and *strictly timid* if the utility function is in $U^{\mathsf{ind}}_{\mathsf{st\text{-}timid}}$.

For $j \in \{1,\ldots,n\}$ and $b \in \{0,1\}$, we write $\mathsf{detect}_{-j} = b$ if $\mathsf{detect}_{j'} = b$ for every $j' \in \{1,\ldots,n\} \setminus \{j\}$. In the analysis of the security of our protocols, we use the following values of utility of adversary $j \in \{1,\ldots,\lambda\}$.

- u_0 is the utility when $\Pr[\mathsf{guess}_j = 1] = \frac{1}{|\mathcal{M}|}$, $\mathsf{suc} = 0$, $\mathsf{detect}_j = 0$, $\mathsf{detect}_{-j} = 1$,
- u_1 is the utility when $\Pr[\mathsf{guess}_j = 1] = \frac{1}{|\mathcal{M}|}$, $\mathsf{suc} = 0$, $\mathsf{detect}_j = 0$, $\mathsf{detect}_{-j} = 0$,
- u_2 is the utility when $\Pr[\mathsf{guess}_j = 1] = \frac{1}{|\mathcal{M}|}$, $\mathsf{suc} = 1$, $\mathsf{detect}_j = 0$, $\mathsf{detect}_{-j} = 0$,

- u_3 is the utility when $\Pr[\mathsf{guess}_j = 1] = \frac{1}{|\mathcal{M}|}$, $\mathsf{suc} = 0$, $\mathsf{detect}_j = 1$, $\mathsf{detect}_{-j} = 0$, and
- u_4 is the utility when $\Pr[\mathsf{guess}_j = 1] = \frac{1}{|\mathcal{M}|}$, $\mathsf{suc} = 1$, $\mathsf{detect}_j = 1$, $\mathsf{detect}_{-j} = 0$.

For any utility function in $U^{\mathsf{ind}}_{\mathsf{timid}}$, it holds that $u_0 > u_1 > \max\{u_2, u_3\}$ and $\min\{u_2, u_3\} > u_4$. If the utility is in $U^{\mathsf{ind}}_{\mathsf{st\text{-}timid}}$, it holds that $u_0 > u_1 > u_2 > u_3 > u_4$.

4 Protocol with Public Channel

We show that the SJST protocol of [30] works as a perfect SMT protocol against independent adversaries. See Sect. A.1 for the description of the protocol. More specifically, we slightly modify the SJST protocol such that in the second and the third rounds, if $b_i = 1$ in B or $v_i = 1$ in V for some $i \in \{1, \ldots, n\}$, the special message "DETECT at i" is also sent together.

Theorem 2. *For any $\lambda \geq 2$, let $t_1, \ldots, t_\lambda$ be integers satisfying $t_1 + \cdots + t_\lambda \leq n$ and $1 \leq t_i \leq n-1$ for every $i \in \{1, \ldots, \lambda\}$. If the parameter ℓ in the SJST protocol satisfies*

$$\ell \geq \max_{t \in \{t_1, \ldots, t_\lambda\}} \left\{ 1 + \log_2 t + \log_2 \frac{u_3 - u_4}{u_2 - u_4 - \alpha}, 1 + \frac{1}{t} \log_2 \frac{u_1 - u_3}{\alpha} \right\}$$

for some $\alpha \in (0, u_2 - u_4)$, then the protocol is perfectly secure against rational $(t_1, \ldots, t_\lambda)$-adversaries with utility function $U \in U^{\mathsf{ind}}_{\mathsf{timid}}$.

Proof. For each $j \in \{1, \ldots, \lambda\}$, let $\mathcal{B}_j$ be the adversary who does not corrupt any channels and outputs a uniformly random message from $\mathcal{M}$ as M_j. Then, the perfect security for $(\mathcal{B}_1, \ldots, \mathcal{B}_\lambda)$ immediately follows.

We show that $(\mathcal{B}_1, \ldots, \mathcal{B}_\lambda)$ is a Nash equilibrium. Since $U_j(\mathcal{B}_1, \ldots, \mathcal{B}_\lambda) = u_2$ for $j \in \{1, \ldots, \lambda\}$, it is sufficient to show that $U_j(\mathcal{A}_j, \mathcal{B}_{-j}) \leq u_2$ for any t_j-adversary $\mathcal{A}_j$. Note that, since the SJST protocol achieves the perfect privacy against at most $n-1$ corruptions, we have that $\Pr[\mathsf{guess}_j = 1] = 1/|\mathcal{M}|$ for any t_j-adversary $\mathcal{A}_j$.

Since messages in the second and the third rounds are sent through the public channel, the adversary $\mathcal{A}_j$ can tamper with messages only in the first round. If $\mathcal{A}_j$ changes the lengths of r_i or R_i of the i-th channel, the tampering will be detected, and hence $\mathsf{detect}_j = 1$. Thus, such tampering cannot increase the utility.

Suppose that $\mathcal{A}_j$ corrupts t_j channels in the first round. Namely, there are exactly t_j distinct i's such that $(r'_i, R'_i) \neq (r_i, R_i)$. Note that a tampering action such that $r'_i \neq r_i$ and $R'_i = R_i$ does not increase the probability that $\mathsf{suc} = 0$, but may only increase that of $\mathsf{detect}_j = 1$. Hence, we assume that $R'_i \neq R_i$ for all the corrupted channels. Also, note that $\mathcal{A}_j$ cannot cause $\mathsf{detect}_{j'}$ for $j' \neq j$ since a message "DETECT at i" is sent only when tampering is made by an adversary who corrupts the i-th channel. Thus, the maximum utility of $U_j(\mathcal{A}_j, \mathcal{B}_{-j})$ is u_1.

We define the following events:

- E_1: No tampering action is detected in the protocol,
- E_2: At least one but not all tampering actions are detected, and
- E_3: All tampering actions are detected.

Note that these three events are disjoint, and either event should occur. Thus, we have that $\Pr[E_1]+\Pr[E_2]+\Pr[E_3]=1$. It follows from the discussion in Sect. A.3 that the probability that the tampering action on one channel is not detected is $2^{1-\ell}$. Since each hash function h_i is chosen independently for each channel i, we have that $\Pr[E_1]=2^{(1-\ell)t_j}$. Similarly, we obtain that $\Pr[E_3]=(1-2^{1-\ell})^{t_j}$. Note that the utility when E_1 occurs is at most u_1. When E_2 occurs, some tampering is detected, but not another tampering. Thus, we have $\mathsf{suc}=0$ and $\mathsf{detect}_j=1$. In the case of E_3, we have $\mathsf{suc}=0$ and $\mathsf{detect}_j=0$. Hence, the utilities when E_2 and E_3 occur are at most u_3 and u_4, respectively. Therefore, the utility of adversary j is

$$\begin{aligned}
U_j(\mathcal{A}_j,\mathcal{B}_{-j}) &\le u_1\cdot\Pr[E_1]+u_3\cdot\Pr[E_2]+u_4\cdot\Pr[E_3]\\
&= u_3+(u_1-u_3)\Pr[E_1]-(u_3-u_4)\Pr[E_3]\\
&\le u_3+(u_1-u_3)\,2^{(1-\ell)t_j}-(u_3-u_4)\left(1-t_j\,2^{1-\ell}\right)\\
&\le u_3+\alpha-(u_3-u_4)\left(1-t_j\,2^{1-\ell}\right) \qquad (1)\\
&\le u_2, \qquad (2)
\end{aligned}$$

where we use the relations $\ell\ge 1+\frac{1}{t_j}\log_2\frac{u_1-u_3}{\alpha}$ and $\ell\ge 1+\log_2 t_j+\log_2\frac{u_3-u_4}{u_2-u_4-\alpha}$ in (1) and (2), respectively. Thus, the utility of adversary j when playing with $(\mathcal{A}_j,\mathcal{B}_{-j})$ is at most u_2 for every $j\in\{1,\dots,\lambda\}$, and hence the statement follows. □

5 Protocol for Minority Corruptions

We provide a non-interactive SMT protocol based on secret-sharing and pairwise independent hash functions. The protocol is secure against independent adversaries who only corrupt minorities of the channels. Namely, we assume that each adversary corrupts at most $\lfloor (n-1)/2\rfloor$ channels. Note that the protocol does not use the public channel as in the protocol in Sect. 4.

We describe the construction of our protocol. The protocol can employ any secret-sharing scheme of threshold $\lfloor (n-1)/2\rfloor$, which may be Shamir's scheme described in Sect. A.2. Let $(s_1,\dots,s_n)$ be the shares generated by the scheme from the message to be sent. Then, pairwise independent hash functions h_i are chosen for each $i\in\{1,\dots,n\}$. For any $j\ne i$, $h_i(s_j)$ is computed as an authentication tag for s_j. Then, $(s_i,h_i,\{h_i(s_j)\}_{j\ne i})$ will be sent through the i-th channel. When s_i is modified to $s_i'\ne s_i$ by some adversary, the modification can be detected by the property of pairwise independent hash functions because the adversary cannot modify all tags $h_j(s_i)$ for $j\ne i$. In addition, a random mask $r_{i,j}$ is applied to $h_i(s_j)$ to conceal the information of s_j in $h_i(s_j)$. The masks $\{r_{j,i}\}_{j\ne i}$ for s_i will be sent through the i-th channel so that only the i-th channel reveals the information of s_i. Hence, the message sent through the i-th channel is $(s_i, h_i, \{h_i(s_j) \oplus r_{i,j}\}_{j \neq i}, \{r_{j,i}\}_{j \neq i})$. As long as minorities of the channels are corrupted by each adversary, a single adversary cannot cause erroneous detection of silent adversaries.

We give a formal description.

Protocol 1. Let $(\mathsf{Share}, \mathsf{Reconst})$ be a secret-sharing scheme of threshold $\lfloor (n-1)/2 \rfloor$, where a secret is chosen from $\mathcal{M}$, and the shares are defined over $\mathcal{V}$. Let $m \in \mathcal{M}$ be the message to be sent by the sender, and $H = \{h \colon \mathcal{V} \to \{0,1\}^\ell\}$ a class of pairwise independent hash functions in Sect. A.3.

1. The sender does the following: Generate the shares $(s_1, \ldots, s_n)$ by $\mathsf{Share}(m)$, and randomly choose $h_i \in H$ for each $i \in \{1, \ldots, n\}$. Also, for every distinct $i, j \in \{1, \ldots, n\}$, choose $r_{i,j} \in \{0,1\}^\ell$ uniformly at random, and then compute $T_{i,j} = h_i(s_j) \oplus r_{i,j}$. Then, for each $i \in \{1, \ldots, n\}$, send $m_i = \big(s_i, h_i, \{T_{i,j}\}_{j \in \{1,\ldots,n\} \setminus \{i\}}, \{r_{j,i}\}_{j \in \{1,\ldots,n\} \setminus \{i\}}\big)$ through the i-th channel.
2. After receiving $\tilde{m}_i = \Big(\tilde{s}_i, \tilde{h}_i, \{\tilde{T}_{i,j}\}_{j \in \{1,\ldots,n\} \setminus \{i\}}, \{\tilde{r}_{j,i}\}_{j \in \{1,\ldots,n\} \setminus \{i\}}\Big)$ on each channel $i \in \{1, \ldots, n\}$, the receiver does the following: For every $i \in \{1, \ldots, n\}$, compute the list $L_i = \Big\{ j \in \{1, \ldots, n\} \colon \tilde{h}_i(\tilde{s}_j) \oplus \tilde{r}_{i,j} \neq \tilde{T}_{i,j} \Big\}$. If a majority of the lists coincide with a list L, reconstruct the message $\tilde{m}$ by $\mathsf{Reconst}(\{i, \tilde{s}_i\}_{i \in \{1,\ldots,n\} \setminus L})$, send messages "DETECT at i" for every $i \in L$, and output $\tilde{m}$.

Theorem 3. *For any $\lambda \geq 2$, let $t_1, \ldots, t_\lambda$ be integers satisfying $t_1 + \cdots + t_\lambda \leq n$ and $1 \leq t_i \leq \lfloor (n-1)/2 \rfloor$ for every $i \in \{1, \ldots, \lambda\}$. If the parameter ℓ in Protocol 1 satisfies*

$$\ell \geq \log_2 \frac{u_1 - u_4}{u_2 - u_4} + 2 \log_2(n+1) - 1,$$

then the protocol is perfectly secure against rational $(t_1, \ldots, t_\lambda)$-adversaries with utility function $U \in U^{\mathsf{ind}}_{\mathsf{timid}}$.

Proof. For $k \in \{1, \ldots, \lambda\}$, let $\mathcal{B}_k$ be the t_k-adversary who does not corrupt any channels and outputs a random message as M_k. First, note that, for any $i \in \{1, \ldots, n\}$, the information of s_i can be obtained only by m_i, the message sent over the i-th channel. This is because for any $j \neq i$, $h_j(s_i)$ is masked as $h_j(s_i) \oplus r_{i,j}$, and the random mask $r_{i,j}$ is included only in m_i. Also, each s_i is a share of the secret sharing of threshold $\lfloor (n-1)/2 \rfloor$. Since $\mathcal{B}_k$ can obtain at most $\lfloor (n-1)/2 \rfloor$ shares, $\mathcal{B}_k$ can learn nothing about the message sent from the sender. Thus, the perfect security is achieved for $(\mathcal{B}_1, \ldots, \mathcal{B}_\lambda)$.

Next, we show that $(\mathcal{B}_1, \ldots, \mathcal{B}_\lambda)$ is a Nash equilibrium. For $k \in \{1, \ldots, \lambda\}$, let $\mathcal{A}_k$ be any t_k-adversary. Since $U_k(\mathcal{B}_1, \ldots, \mathcal{B}_\lambda) = u_2$, to increase the utility, $\mathcal{A}_k$ needs to get either (a) $\mathsf{suc} = 0$, or (b) $\mathsf{suc} = 1$, $\mathsf{detect}_k = 0$, and $\mathsf{detect}_{k'} = 1$ for some $k' \neq k$.

For the case of (a), $\mathcal{A}_k$ tries to change s_i into $\tilde{s}_i \neq s_i$ for some $i \in \{1, \ldots, n\}$. Since $\mathcal{A}_k$ does not corrupt some $i' \in \{1, \ldots, n\}$, the index i corrupted by $\mathcal{A}_k$ will be included in the list $L_{i'}$ unless $h_{i'}(\tilde{s}_i) \oplus \tilde{r}_{i',i} = T_{i',i}$. Note that $\tilde{s}_i$ and $\tilde{r}_{i',i}$ are included in $\tilde{m}_i$, and thus can be changed, but $h_{i'}$ and $T_{i',i}$ are in $\tilde{m}_{i'}$, and thus have been unchanged. It follows from the property of pairwise independent hash functions that this can happen with probability $2^{1-\ell}$ assuming $\tilde{s}_i \neq s_i$. Thus, i will be included in $L_{i'}$ with probability at least $1-2^{1-\ell}$. Since there are at least $n - \lfloor (n-1)/2 \rfloor = \lceil (n+1)/2 \rceil$ such indices i', the probability that a majority of the lists contains i is at least $1 - \lceil (n+1)/2 \rceil \cdot 2^{1-\ell}$. Note that $\mathcal{A}_k$ may corrupt $\lfloor (n-1)/2 \rfloor$ channels in total. The probability that all the corrupted indices coincide with a majority of the list is at least $1 - \lfloor (n-1)/2 \rfloor \cdot \lceil (n+1)/2 \rceil \cdot 2^{1-\ell} \geq 1 - (n+1)^2 \cdot 2^{-(\ell+1)}$. In that case, the message can be reconstructed by other shares, and thus we have $\mathsf{suc} = 1$, $\mathsf{detect}_k = 1$, and $\mathsf{detect}_{k'} = 0$ for $k' \neq k$, resulting in the utility of u_4. Since $\mathcal{A}_k$ only corrupts a minority of the channels, it cannot cause $\mathsf{detect}_{k'} = 1$ for $k' \neq k$. Thus, the maximum utility of $\mathcal{A}_k$ is u_1. Thus, the utility of adversary k when tampering as $\tilde{s}_i \neq s_i$ is at most

$$U_k(\mathcal{A}_k, \mathcal{B}_{-k}) \leq (n+1)^2 \cdot 2^{-(\ell+1)} \cdot u_1 + \left(1 - (n+1)^2 \cdot 2^{-(\ell+1)}\right) \cdot u_4,$$

which is at most u_2 by the assumption on ℓ.

For the case of (b), $\mathcal{A}_k$ needs to generate the corrupted message $\tilde{m}_i$ for the i-th channel so that for a majority of indices $j \in \{1, \ldots, n\}$, $\tilde{h}_i(s_j) \oplus r_{i,j} \neq \tilde{T}_{i,j}$, where each j is corrupted by $\mathcal{B}_{k'}$ with $k' \neq k$, and thus $r_{i,j}$ and s_j are not tampered with. Since $\mathcal{A}_k$ only corrupts a minority of the channels, this cannot happen.

Therefore, $(\mathcal{B}_1, \ldots, \mathcal{B}_\lambda)$ is a Nash equilibrium. □

6 Protocol for Majority Corruptions

We present a protocol against adversaries who may corrupt a majority of the channels. We assume that adversaries are *strictly* timid in this setting. The protocol is a minor modification of the protocol for minority corruptions. In Protocol 1, the lists L_i's of the corrupted channels are generated for each channel, and the final list L is determined by the majority voting. Thus, if an adversary corrupts a majority of the channels, the result of the majority voting can be easily forged, and hence the protocol does not work for majority corruption.

To cope with majority corruptions, we modify the protocol such that (1) the threshold of the secret sharing is changed from $\lfloor (n-1)/2 \rfloor$ to $n-1$, and (2) the final list L of the corrupted channels is composed of the union of all the set L_i, namely, $L = L_1 \cup \cdots \cup L_n$. The threshold of $n-1$ can be achieved by Shamir's scheme. Intuitively, this protocol works for strictly timid adversaries because any tampering detection is approved without voting and thus such adversaries will keep silent not to be detected.

We give a formal description of the protocol.

Protocol 2. Let $(\mathsf{Share}, \mathsf{Reconst})$ be a secret-sharing scheme of threshold $n-1$, where a secret is chosen from $\mathcal{M}$, and the shares are defined over $\mathcal{V}$. Let $m \in \mathcal{M}$ be the message to be sent by the sender, and $H = \{h\colon \mathcal{V} \to \{0,1\}^\ell\}$ a class of pairwise independent hash functions in Sect. A.3.

1. The sender does the following: Generate the shares $(s_1, \ldots, s_n)$ by $\mathsf{Share}(m)$, and randomly choose $h_i \in H$ for each $i \in \{1, \ldots, n\}$. Also, for every distinct $i, j \in \{1, \ldots, n\}$, choose $r_{i,j} \in \{0,1\}^\ell$ uniformly at random, and then compute $T_{i,j} = h_i(s_j) \oplus r_{i,j}$. Then, for each $i \in \{1, \ldots, n\}$, send $m_i = \left(s_i, h_i, \{T_{i,j}\}_{j \in \{1,\ldots,n\} \setminus \{i\}}, \{r_{j,i}\}_{j \in \{1,\ldots,n\} \setminus \{i\}}\right)$ through the i-th channel.
2. After receiving $\tilde{m}_i = \left(\tilde{s}_i, \tilde{h}_i, \{\tilde{T}_{i,j}\}_{j \in \{1,\ldots,n\} \setminus \{i\}}, \{\tilde{r}_{j,i}\}_{j \in \{1,\ldots,n\} \setminus \{i\}}\right)$ on each channel $i \in \{1, \ldots, n\}$, the receiver does the following: For every $i \in \{1, \ldots, n\}$, compute the list $L_i = \left\{j \in \{1, \ldots, n\} \colon \tilde{h}_i(\tilde{s}_j) \oplus \tilde{r}_{i,j} \neq \tilde{T}_{i,j}\right\}$. Then, set $L = L_1 \cup \cdots \cup L_n$. If $L = \emptyset$, reconstruct the message $\tilde{m}$ by $\mathsf{Reconst}(\{i, \tilde{s}_i\}_{i \in \{1,\ldots,n\}})$, and output $\tilde{m}$. Otherwise, send messages "DETECT at i" for every $i \in L$, and output $\perp$ as the failure symbol.

Theorem 4. *For any $\lambda \geq 2$, let $t_1, \ldots, t_\lambda$ be integers satisfying $t_1 + \cdots + t_\lambda \leq n$ and $1 \leq t_i \leq n-1$ for every $i \in \{1, \ldots, \lambda\}$. If the parameter ℓ in Protocol 2 satisfies*

$$\ell \geq \log_2 \frac{u_0 - u_3}{u_2 - u_3} - 1,$$

then the protocol is perfectly secure against rational $(t_1, \ldots, t_\lambda)$-adversaries with utility function $U \in U^{\mathsf{ind}}_{\mathsf{st\text{-}timid}}$.

Proof. For $k \in \{1, \ldots, \lambda\}$, we define $\mathcal{B}_k$ as the t_k-adversary who does not corrupt any channels and outputs a random message as M_j. By the same reason as in the proof of Theorem 3, the protocol is perfectly secure against $(\mathcal{B}_1, \ldots, \mathcal{B}_\lambda)$.

Next, we show that $(\mathcal{B}_1, \ldots, \mathcal{B}_\lambda)$ is a Nash equilibrium. Let $\mathcal{A}_k$ be any t_k-adversary for $k \in \{1, \ldots, \lambda\}$. As in the proof of Theorem 3, $\mathcal{A}_k$ needs to yield either (a) $\mathsf{suc} = 0$, or (b) $\mathsf{suc} = 1$, $\mathsf{detect}_k = 0$, and $\mathsf{detect}_{k'} = 1$ for some $k' \neq k$. For the case of (a), $\mathcal{A}_k$ needs to corrupt the i-th channel so that $\tilde{s}_i \neq s_i$. Since there is at least one index $i' \in \{1, \ldots, n\}$ that is corrupted by $\mathcal{B}_{k'}$ with $k' \neq k$, the index i is included in the list $L_{i'}$ with probability at least $1 - 2^{1-\ell}$. Thus, the utility of adversary k is at most

$$U_k(\mathcal{A}_k, \mathcal{B}_{-k}) \leq 2^{-(\ell+1)} \cdot u_0 + \left(1 - 2^{-(\ell+1)}\right) \cdot u_3,$$

which is at most u_2 by assumption. For the case of (b), if some index is in the final list L, since the threshold of secret sharing is $n-1$, the message is not reconstructed. Then we have $\mathsf{suc} = 0$. Namely, (b) does not happen. Thus, $(\mathcal{B}_1, \ldots, \mathcal{B}_\lambda)$ is a Nash equilibrium. □

7 SMT Against Malicious and Rational Adversaries

In the previous sections, we have discussed SMT against independent rational adversaries. We have assumed that all the adversaries behave rationally. The assumption may be strong in the sense that all of them can be characterized by the utility function we defined. In this section, we discuss more realistic situations in which some adversary may not behave rationally, but maliciously.

7.1 Rational SMT in the Presence of a Malicious Adversary

Without loss of generality, we assume that there are $\lambda \geq 2$ adversaries, and adversaries $1, \ldots, \lambda - 1$ are rational, and adversary λ behaves maliciously. We use the same definitions of the SMT game and the utility function in Sect. 3. We define *robust* security against rational adversaries. A similar definition appeared in the context of rational secret sharing [1]. For strategies $\mathcal{B}_1, \ldots, \mathcal{B}_{\lambda-1}, \mathcal{A}_\lambda, \mathcal{A}_j$ for $j \in \{1, \ldots, \lambda - 1\}$, we denote by $(\mathcal{A}_j, \mathcal{B}_{-j}, \mathcal{A}_\lambda)$ the strategy profile $(\mathcal{B}_1, \ldots, \mathcal{B}_{j-1}, \mathcal{A}_j, \mathcal{B}_{j+1}, \ldots, \mathcal{B}_{\lambda-1}, \mathcal{A}_\lambda)$.

Definition 4 (Security of Robust RSMT). *An SMT protocol Π is* t^*-robust perfectly secure against rational $(t_1, \ldots, t_{\lambda-1})$-adversaries with utility function U *if there are t_j-adversary $\mathcal{B}_j$ for $j \in \{1, \ldots, \lambda - 1\}$ such that for any t_j-adversary $\mathcal{A}_j$ for $j \in \{1, \ldots, \lambda - 1\}$ and t^*-adversary $\mathcal{A}_\lambda$,*

1. Perfect security*: Π is $(0, 0)$-SMT against $(\mathcal{B}_1, \ldots, \mathcal{B}_{\lambda-1}, \mathcal{A}_\lambda)$, and*
2. Robust Nash equilibrium*: $U_j(\mathcal{A}_j, \mathcal{B}_{-j}, \mathcal{A}_\lambda, U) \leq U_j(\mathcal{B}_j, \mathcal{B}_{-j}, \mathcal{A}_\lambda, U)$ for every $j \in \{1, \ldots, \lambda - 1\}$ in the SMT game.*

Compared to Definition 3, robust RSMT requires that the perfect security is achieved even in the presence of a malicious adversary $\mathcal{A}_\lambda$, and a strategy profile $(\mathcal{B}_1, \ldots, \mathcal{B}_{\lambda-1}, \mathcal{A}_\lambda)$ is a Nash equilibrium for adversary $j \in \{1, \ldots, \lambda - 1\}$.

7.2 Protocol Against Malicious and Rational Adversaries

We show that a robust RSMT protocol can be constructed based on the protocol for minority corruption in Sect. 5. For t^*-robust against $(t_1, \ldots, t_{\lambda-1})$-adversaries, we assume that $t^* \leq \lfloor (n-1)/3 \rfloor$ and $1 \leq t_j \leq \min\{\lfloor (n-1)/2 \rfloor - t^*, (n-1)/3 \rfloor\}$ for each $j \in \{1, \ldots, \lambda - 1\}$. Our non-interactive protocol is obtained simply by modifying the threshold of the secret sharing in Protocol 1 from $\lfloor (n-1)/2 \rfloor$ to $\lfloor (n-1)/3 \rfloor$. This protocol works because when only a malicious adversary corrupts at most $\lfloor (n-1)/3 \rfloor$ channels, the transmission failure does not occur due to the error-correction property of the secret sharing. Thus, perfect security is achieved in the presence of a malicious adversary. Even if some rational adversary deviates from the protocol together with a malicious adversary, they can affect at most $t_j + t^* \leq \lfloor (n-1)/2 \rfloor$ votes, and thus any tampering will be identified with high probability by the majority voting.

The formal description is given below.

Protocol 3. Let $(\mathsf{Share}, \mathsf{Reconst})$ be a secret-sharing scheme of threshold $\lfloor (n-1)/3 \rfloor$, where a secret is chosen from $\mathcal{M}$, the shares are defined over $\mathcal{V}$, and the secret can be reconstructed even if $\lfloor (n-1)/3 \rfloor$ out of n shares are tampered with. Let $m \in \mathcal{M}$ be the message to be sent by the sender, and $H = \{h \colon \mathcal{V} \to \{0,1\}^\ell\}$ a class of pairwise independent hash functions in Sect. A.3.

1. The sender does the following: Generate the shares $(s_1, \ldots, s_n)$ by $\mathsf{Share}(m)$, and randomly choose $h_i \in H$ for each $i \in \{1, \ldots, n\}$. For every distinct $i, j \in \{1, \ldots, n\}$, choose $r_{i,j} \in \{0,1\}^\ell$ uniformly at random, and

then compute $T_{i,j} = h_i(s_j) \oplus r_{i,j}$. For each $i \in \{1, \ldots, n\}$, send $m_i = \left(s_i, h_i, \{T_{i,j}\}_{j \in \{1,\ldots,n\} \setminus \{i\}}, \{r_{j,i}\}_{j \in \{1,\ldots,n\} \setminus \{i\}}\right)$ through the i-th channel.

2. After receiving $\tilde{m}_i = \left(\tilde{s}_i, \tilde{h}_i, \{\tilde{T}_{i,j}\}_{j \in \{1,\ldots,n\} \setminus \{i\}}, \{\tilde{r}_{j,i}\}_{j \in \{1,\ldots,n\} \setminus \{i\}}\right)$ on each channel $i \in \{1, \ldots, n\}$, the receiver does the following: For every $i \in \{1, \ldots, n\}$, compute the list $L_i = \left\{j \in \{1, \ldots, n\} \colon \tilde{h}_i(\tilde{s}_j) \oplus \tilde{r}_{i,j} \neq \tilde{T}_{i,j}\right\}$. If a majority of the list coincide with a list L, reconstruct the message $\tilde{m}$ by $\mathsf{Reconst}(\{i, \tilde{s}_i\}_{i \in \{1,\ldots,n\}})$, send message "DETECT at i" for every $i \in L$, and output $\tilde{m}$.

For the security analysis, we define the values of utility of adversary $j \in \{1, \ldots, \lambda - 1\}$ such that

- u_1' is the utility in the same case as u_1 except that $\mathsf{detect}_\lambda = 1$,
- u_2' is the utility in the same case as u_2 except that $\mathsf{detect}_\lambda = 1$, and
- u_4' is the utility in the same case as u_4 except that $\mathsf{detect}_\lambda = 1$.

The values u_1, u_2, u_4 are defined as the case that $\mathsf{detect}_{j'} = 0$ for every $j' \in \{1, \ldots, \lambda\} \setminus \{j\}$. In the above, the values u_1', u_2', u_4' are defined as $\mathsf{detect}_{j'} = 0$ for every $j' \in \{1, \ldots, \lambda - 1\} \setminus \{j\}$ and $\mathsf{detect}_\lambda = 1$.

Theorem 5. *For any $\lambda \geq 2$, let $t_1, \ldots, t_{\lambda-1}, t^*$ be integers satisfying $t_1 + \cdots + t_{\lambda-1} + t^* \leq n$, $0 \leq t^* \leq \lfloor (n-1)/3 \rfloor$, and $1 \leq t_i \leq \min\{\lfloor (n-1)/2 \rfloor - t^*, \lfloor (n-1)/3 \rfloor\}$ for every $i \in \{1, \ldots, \lambda - 1\}$. If the parameter ℓ in Protocol 3 satisfies*

$$\ell \geq \max_{(u_1^*, u_2^*, u_4^*) \in \{(u_1, u_2, u_4), (u_1', u_2', u_4')\}} \left\{ \log_2 \frac{u_1^* - u_4^*}{u_2^* - u_4^*} + 2 \log_2 (n+1) - 1 \right\},$$

then the protocol is t^*-robust perfectly secure against rational $(t_1, \ldots, t_{\lambda-1})$-adversaries with utility function $U \in U_{\mathsf{timid}}^{\mathsf{ind}}$.

Proof. For $k \in \{1, \ldots, \lambda - 1\}$, let $\mathcal{B}_k$ be the t_k-adversary who does not corrupt any channels, and output a random message as M_k. Let $\mathcal{A}_\lambda$ be any t^*-adversary. Note that the information of s_i can be obtained only by seeing m_i since each $h_j(s_i)$ is masked by $r_{j,i}$, which is included only in m_i. Since each s_i is a share of the secret sharing of threshold $\lfloor (n-1)/3 \rfloor$, each adversary $\mathcal{B}_k$ and $\mathcal{A}_\lambda$ can learn nothing about the original message. Although at most t^* messages may be corrupted by $\mathcal{A}_\lambda$, it follows from the property of the underlying secret sharing that the message can be correctly recovered in the presence of $t^* \leq \lfloor (n-1)/3 \rfloor$ corruptions out of n shares. Thus, the protocol is perfectly secure against $(\mathcal{B}_1, \ldots, \mathcal{B}_{\lambda-1}, \mathcal{A}_\lambda)$.

Next, we show that $(\mathcal{B}_1, \ldots, \mathcal{B}_{\lambda-1}, \mathcal{A}_\lambda)$ is a Nash equilibrium for any $\mathcal{A}_\lambda$. When the strategy profile $(\mathcal{B}_1, \ldots, \mathcal{B}_{\lambda-1}, \mathcal{A}_\lambda)$ is employed, we have $\mathsf{suc} = 1$. Hence, to increase the utility of adversary k, $\mathcal{A}_k$ needs to get either (a) $\mathsf{suc} = 0$, or (b) $\mathsf{suc} = 1$, $\mathsf{detect}_k = 0$, and $\mathsf{detect}_{k'} = 1$ for some $k' \neq k$.

For the case of (a), $\mathcal{A}_k$ tries to change s_i into $\tilde{s}_i \neq s_i$ for some $i \in \{1, \ldots, n\}$. When playing with $(\mathcal{A}_k, \mathcal{B}_{-k}, \mathcal{A}_\lambda)$, the number of corrupted channels is at most $t_k + t^* \leq \lfloor (n-1)/2 \rfloor$. Hence, there are a majority of indices i' that is not corrupted by $\mathcal{A}_k$ or $\mathcal{A}_\lambda$, and for each i', the tampering on the i-th channel will be detected, namely, the list $L_{i'}$ will include i with high probability. By the same argument as in the proof of Theorem 3, any tampering of $\tilde{s}_i \neq s_i$ by $\mathcal{A}_k$ and $\mathcal{A}_\lambda$ is detected with probability at least $1 - (n+1)^2 \cdot 2^{-(\ell+1)}$. Thus, we have that

$$U_k(\mathcal{A}_k, \mathcal{B}_{-k}, \mathcal{A}_\lambda) \leq (n+1)^2 \cdot 2^{-(\ell+1)} \cdot u_1^* + \left(1 - (n+1)^2 \cdot 2^{-(\ell+1)}\right) \cdot u_4^*,$$

which is at most u_2^* by the assumption on ℓ.

For the case of (b), $\mathcal{A}_k$ needs the result that $j \in L_i$ for a majority of the list L_i's, where the j-th channel is corrupted by adversary k'. However, since $\mathcal{A}_k$ and $\mathcal{A}_\lambda$ can corrupt a minority of the channels, this event cannot happen.

Thus, we have shown that $(\mathcal{B}_1, \ldots, \mathcal{B}_{\lambda-1})$ is a robust Nash equilibrium. □

8 Conclusions

We have studied the problem of constructing SMT protocols against adversaries who may corrupt all the channels between the sender and the receiver. If all adversaries are malicious, we cannot hope for reliable transmission because adversaries who interrupt all the messages can cause transmission failure. Also, if a single adversary corrupts all the channels, we cannot achieve privacy since the adversary can obtain the same information as the receiver who can recover the transmitted message. We show that if multiple rational adversaries exclusively corrupt the channels, perfectly secure SMT protocols can be constructed. Our results demonstrate that even if all the physical resources may be corrupted by adversaries, it is possible to provide secure protocols by considering the rationality and independence of each group of adversaries.

Acknowledgments. This work was supported in part by JSPS Grants-in-Aid for Scientific Research Numbers 16H01705, 17H01695, 18K11159, and 19K22849.

A Building Blocks

A.1 The SJST protocol

We describe an almost-reliable SMT protocol using the public channel proposed by Shi, Jiang, Safavi-Naini, and Tuhin [30]. We refer it as the SJST protocol.

The protocol is based on the simple protocol for "static" adversaries in which the sender sends a random key R_i over the i-th channel for each $i \in \{1, \ldots, n\}$, and the encrypted message $c = m \oplus R_1 \oplus \cdots \oplus R_n$ over the public channel. Suppose that the adversary sees the messages sent over the corrupted channels, but does not change them. Since the adversary cannot see at least one key R_j when corrupting less than n channels, the mask $R_1 \oplus \cdots \oplus R_n$ for the encryption looks random for the adversary. Thus, the message m can be securely encrypted and reliably sent through the public channel. To cope with "active" adversaries, who may change messages sent over the corrupted channels, the SJST protocol employs a mechanism for detecting the adversary's tampering by using hash functions. Specifically, the *pairwise independent* hash functions (see Sect. A.3) satisfy the following property: when a pair of keys (r_i, R_i) is changed to $(r'_i, R'_i) \neq (r_i, R_i)$, the hash value for (r_i, R_i) is different from that for (r'_i, R'_i) with high probability if the hash function is chosen randomly after the tampering occurred. In the SJST protocol, the sender sends a pair of keys (r_i, R_i) over the i-th channel. Then, the receiver chooses n pairwise independent hash functions h_i's, and sends them over the public channel. By comparing hash values for (r_i, R_i)'s sent by the sender with those for (r'_i, R'_i)'s received by the receiver, they can identify the channels for which messages, i.e., keys, were tampered with. By ignoring keys sent over such channels, the sender can correctly encrypt a message m with untampered keys and send the encryption reliably over the public channel.

We describe the SJST protocol below, which is a three-round protocol, and achieves the reliability with $\delta = (n-1) \cdot 2^{1-\ell}$, where ℓ is the length of hash values.

Protocol 4 (The SJST protocol [30]). Let n be the number of channels, $m \in \mathcal{M}$ the message to be sent by the sender $\mathcal{S}$, and $H = \{h\colon \{0,1\}^k \to \{0,1\}^\ell\}$ a class of pairwise independent hash functions.

1. For each $i \in \{1, \dots, n\}$, $\mathcal{S}$ chooses $r_i \in \{0,1\}^\ell$ and $R_i \in \{0,1\}^k$ uniformly at random, and sends the pair (r_i, R_i) over the i-th channel.
2. For each $i \in \{1, \dots, n\}$, $\mathcal{R}$ receives (r'_i, R'_i) through the i-th channel, and then chooses $h_i \leftarrow H$ uniformly at random. If $|r'_i| \neq \ell$ or $|R'_i| \neq k$, set $b_i = 1$, and otherwise, set $b_i = 0$. Then, set $T'_i = r'_i \oplus h_i(R'_i)$, and $H_i = (h_i, T'_i)$ if $b_i = 0$, and $H_i = \perp$ otherwise. Finally, $\mathcal{R}$ sends $(B, H_1, \dots, H_n)$ over the public channel, where $B = (b_1, \dots, b_n)$.
3. $\mathcal{S}$ receives $(B, H_1, \dots, H_n)$ through the public channel. For each $i \in \{1, \dots, n\}$ with $b_i = 0$, $\mathcal{S}$ computes $T_i = r_i \oplus h_i(R_i)$, and sets $v_i = 0$ if $T_i = T'_i$, and $v_i = 1$ otherwise. Then, $\mathcal{S}$ sends (V, c) over the public channel, where $V = (v_1, \dots, v_n)$, and $c = m \oplus (\bigoplus_{v_i=0} R_i)$.
4. On receiving (V, c), $\mathcal{R}$ recovers $m = c \oplus (\bigoplus_{v_i=0} R_i)$.

Theorem 6 ([30]). *The SJST protocol is* $(0, (n-1) \cdot 2^{1-\ell})$*-SMT against* t*-adversary for any* $t < n$.

We can find a complete proof of the above theorem in [30]. For self-containment, we give a brief sketch of the proof.

- *Privacy*: The adversary can get $c = m \oplus (\bigoplus_{v_i=0} R_i)$ through the public channel. Since m is masked by uniformly random R_i's, the adversary has to corrupt all the i-th channels with $v_i = 0$ to recover m. However, since any t-adversary can corrupt at most t $(< n)$ channels, the adversary can cause $v_i = 1$ for at most $n - 1$ i's. Hence, there is at least one i with $v_i = 0$, for which the adversary cannot obtain R_i. Thus, the protocol satisfies the perfect privacy.

- *Reliability*: Since the protocol uses the public channel at the second and the third rounds, the adversary can tamper with channels only at the first round. Suppose that the adversary tampers with (r_i, R_i). If $R_i \neq R'_i$ and $T_i = T'_i$, then $\mathcal{R}$ would recover a wrong message, but the tampering is not detected. It follows from the property of pairwise independent hash functions (see Sect. A.3) that the probability that the above event happens is at most $(n-1)2^{1-\ell}$. Thus, the protocol achieves the reliability with $\delta = (n-1) \cdot 2^{1-\ell}$.

A.2 Secret Sharing

Secret sharing, introduced by Shamir [29] and Blackley [8], enables us to distribute the secret information securely. Let $s \in \mathbb{F}$ be a secret from some finite field $\mathbb{F}$. A (threshold) secret-sharing scheme provides a way for distributing s into n shares $s_1, \ldots, s_n$ such that, for some parameter $t > 0$, (1) any t shares give no information about s, and (2) any $t + 1$ shares uniquely determine s.

Definition 5. *Let t, n be positive integers with $t < n$. A (t, n)*-secret sharing scheme *with range $\mathcal{G}$ consists of two algorithms* $(\mathsf{Share}, \mathsf{Reconst})$ *satisfying the following conditions:*

- Correctness*: For any $s \in \mathcal{G}$ and $I \subseteq \{1, \ldots, n\}$ with $|I| > t$,*

$$\Pr\left[(\tilde{s}, J) \leftarrow \mathsf{Reconst}\left(\{i, s_i\}_{i \in I}\right) \wedge \tilde{s} = s\right] = 1,$$

where $(s_1, \ldots, s_n) \leftarrow \mathsf{Share}(s)$, and

- Perfect Privacy*: For any $s, s' \in \mathcal{G}$ and $I \subseteq \{1, \ldots, n\}$ with $|I| \leq t$,*

$$\mathrm{SD}\left(\{s_i\}_{i \in I}, \{s'_i\}_{i \in I}\right) = 0,$$

where $(s_1, \ldots, s_n) \leftarrow \mathsf{Share}(s)$ and $(s'_1, \ldots, s'_n) \leftarrow \mathsf{Share}(s')$.

Shamir [29] gave a (t, n)-secret sharing scheme based on polynomial evaluations for any $t < n$. Let $\mathbb{F}$ be a finite field of size at least n. Then, for a given secret $s \in \mathbb{F}$, the sharing algorithm chooses random elements $r_1, \ldots, r_t \in \mathbb{F}$, and constructs a polynomial $f(x) = s + r_1 x + r_2 x^2 + \cdots + r_t x^t$ of degree t over $\mathbb{F}$. Then, for a fixed set of n distinct elements $\{a_1, \ldots, a_n\} \subseteq \mathbb{F}$, the i-th share is $f(a_i)$ for $i \in \{1, \ldots, n\}$. Given $\{i, f(a_i)\}_{i \in I}$ for $|I| > t$, the reconstruction algorithm recovers the polynomial f by polynomial interpolation, and outputs $f(0) = s$ as a recovered secret.

McEliece and Sarwate [26] observed that Shamir's scheme is closely related to Reed-Solomon codes, and thus the shares can be efficiently recovered even if some of them have been tampered with. We will use the useful fact that even if at most $\lfloor (n-1)/3 \rfloor$ out of the n shares are tampered with, the original secret can be correctly recovered by decoding algorithms of Reed-Solomon codes.

A.3 Pairwise Independent Hash Functions

Wegman and Carter [33] introduced the notion of pairwise independent (or strongly universal) hash functions and gave its construction. As in the SJST protocol described above, our protocols employ pairwise independent hash functions.

Definition 6. *Suppose that a class of hash functions* $H = \{h\colon \{0,1\}^m \to \{0,1\}^\ell\}$*, where* $m \geq \ell$*, satisfies the following: for any distinct* $x_1, x_2 \in \{0,1\}^m$ *and* $y_1, y_2 \in \{0,1\}^\ell$*,*

$$\Pr_{h \in H}[h(x_1) = y_1 \wedge h(x_2) = y_2] \leq \gamma.$$

Then H *is called* γ-pairwise independent*. In the above, the randomness comes from the uniform choice of* h *over* H*.*

Here we mention a useful property of almost pairwise independent hash function, which guarantees the security of some SMT protocols.

Lemma 1 ([30]). *Let* $H = \{h\colon \{0,1\}^m \to \{0,1\}^\ell\}$ *be a* γ*-almost pairwise independent hash function family. Then for any* $(x_1, c_1) \neq (x_2, c_2) \in \{0,1\}^m \times \{0,1\}^\ell$*, we have*

$$\Pr_{h \in H}[c_1 \oplus h(x_1) = c_2 \oplus h(x_2)] \leq 2^\ell \gamma.$$

In [33], Wegman and Carter constructed a family of $2^{1-2\ell}$-almost pairwise independent hash functions. In particular, their hash function family $H_{wc} = \{h\colon \{0,1\}^m \to \{0,1\}^\ell\}$ satisfies that

$$\Pr_{h \in H_{wc}}[h(x_1) = y_1 \wedge h(x_2) = y_2] = 2^{1-2\ell}$$

for any distinct $x_1, x_2 \in \{0,1\}^m$ and for any $y_1, y_2 \in \{0,1\}^\ell$ and also

$$\Pr_{h \in H_{wc}}[c_1 \oplus h(x_1) = c_2 \oplus h(x_2)] = 2^{1-\ell} \tag{3}$$

for any distinct pairs $(x_1, c_1) \neq (x_2, c_2) \in \{0,1\}^m \times \{0,1\}^\ell$.

References

1. Abraham, I., Dolev, D., Gonen, R., Halpern, J.Y.: Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In: Ruppert, E., Malkhi, D. (eds.) PODC, pp. 53–62. ACM (2006)
2. Abraham, I., Dolev, D., Halpern, J.Y.: Distributed protocols for leader election: a game-theoretic perspective. In: Afek, Y. (ed.) DISC 2013. LNCS, vol. 8205, pp. 61–75. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41527-2_5
3. Agarwal, S., Cramer, R., de Haan, R.: Asymptotically optimal two-round perfectly secure message transmission. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 394–408. Springer, Heidelberg (2006). https://doi.org/10.1007/11818175_24

4. Asharov, G., Canetti, R., Hazay, C.: Towards a game theoretic view of secure computation. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 426–445. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_24
5. Asharov, G., Canetti, R., Hazay, C.: Toward a game theoretic view of secure computation. J. Cryptol. **29**(4), 879–926 (2016)
6. Asharov, G., Lindell, Y.: Utility dependence in correct and fair rational secret sharing. J. Cryptol. **24**(1), 157–202 (2011)
7. Azar, P.D., Micali, S.: Super-efficient rational proofs. In: Kearns, M., McAfee, R.P., Tardos,É. (eds.) EC 2013, pp. 29–30. ACM (2013)
8. Blakley, G.R.: Safeguarding cryptographic keys. In: Proceedings of the National Computer Conference 1979, vol. 48, pp. 313–317 (1979)
9. Dolev, D., Dwork, C., Waarts, O., Yung, M.: Perfectly secure message transmission. J. ACM **40**(1), 17–47 (1993)
10. Franklin, M.K., Wright, R.N.: Secure communication in minimal connectivity models. J. Cryptol. **13**(1), 9–30 (2000)
11. Fuchsbauer, G., Katz, J., Naccache, D.: Efficient rational secret sharing in standard communication networks. In: Micciancio [27], pp. 419–436
12. Fujita, M., Yasunaga, K., Koshiba, T.: Perfectly secure message transmission against rational timid adversaries. In: Bushnell, L., Poovendran, R., Başar, T. (eds.) GameSec 2018. LNCS, vol. 11199, pp. 127–144. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01554-1_8
13. Garay, J.A., Katz, J., Maurer, U., Tackmann, B., Zikas, V.: Rational protocol design: cryptography against incentive-driven adversaries. In: FOCS, pp. 648–657. IEEE Computer Society (2013)
14. Garay, J.A., Katz, J., Tackmann, B., Zikas, V.: How fair is your protocol?: a utility-based approach to protocol optimality. In: Georgiou, C., Spirakis, P.G. (eds.) PODC, pp. 281–290. ACM (2015)
15. Garay, J.A., Ostrovsky, R.: Almost-everywhere secure computation. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 307–323. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_18
16. Gradwohl, R.: Rationality in the full-information model. In: Micciancio [27], pp. 401–418
17. Groce, A., Katz, J.: Fair computation with rational players. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 81–98. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_7
18. Groce, A., Katz, J., Thiruvengadam, A., Zikas, V.: Byzantine agreement with a rational adversary. In: Czumaj, A., Mehlhorn, K., Pitts, A., Wattenhofer, R. (eds.) ICALP 2012. LNCS, vol. 7392, pp. 561–572. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31585-5_50
19. Guo, S., Hubácek, P., Rosen, A., Vald, M.: Rational arguments: single round delegation with sublinear verification. In: Naor, M. (ed.) ITCS, pp. 523–540. ACM (2014)
20. Halpern, J.Y., Teague, V.: Rational secret sharing and multiparty computation: extended abstract. In: Babai, L. (ed.) Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, 13–16 June 2004, pp. 623–632. ACM (2004)
21. Halpern, J.Y., Vilaça, X.: Rational consensus: extended abstract. In: Giakkoupis, G. (ed.) PODC, pp. 137–146. ACM (2016)

22. Hayashi, M., Koshiba, T.: Universal construction of cheater-identifiable secret sharing against rushing cheaters based on message authentication. In: 2018 IEEE International Symposium on Information Theory, ISIT 2018, Vail, CO, USA, 17–22 June 2018, pp. 2614–2618. IEEE (2018)
23. Inasawa, K., Yasunaga, K.: Rational proofs against rational verifiers. IEICE Trans. **100-A**(11), 2392–2397 (2017)
24. Kawachi, A., Okamoto, Y., Tanaka, K., Yasunaga, K.: General constructions of rational secret sharing with expected constant-round reconstruction. Comput. J. **60**(5), 711–728 (2017)
25. Kurosawa, K., Suzuki, K.: Truly efficient 2-round perfectly secure message transmission scheme. IEEE Trans. Inf. Theory **55**(11), 5223–5232 (2009)
26. McEliece, R.J., Sarwate, D.V.: On sharing secrets and reed-solomon codes. Commun. ACM **24**(9), 583–584 (1981)
27. Micciancio, D. (ed.): TCC 2010. LNCS, vol. 5978. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2
28. Sayeed, H.M., Abu-Amara, H.: Efficient perfectly secure message transmission in synchronous networks. Inf. Comput. **126**(1), 53–61 (1996)
29. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
30. Shi, H., Jiang, S., Safavi-Naini, R., Tuhin, M.A.: On optimal secure message transmission by public discussion. IEEE Trans. Inf. Theory **57**(1), 572–585 (2011)
31. Spini, G., Zémor, G.: Perfectly secure message transmission in two rounds. In: Hirt, M., Smith, A. (eds.) TCC 2016. LNCS, vol. 9985, pp. 286–304. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53641-4_12
32. Srinathan, K., Narayanan, A., Rangan, C.P.: Optimal perfectly secure message transmission. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 545–561. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28628-8_33
33. Wegman, M.N., Carter, L.: New hash functions and their use in authentication and set equality. J. Comput. Syst. Sci. **22**(3), 265–279 (1981)
34. Yasunaga, K.: Public-key encryption with lazy parties. IEICE Trans. **99-A**(2), 590–600 (2016)
35. Yasunaga, K., Yuzawa, K.: Repeated games for generating randomness in encryption. IEICE Trans. **101-A**(4), 697–703 (2018)