Perfectly Secure Message Transmission against Rational Adversaries[*]

Maiki Fujita[†] Takeshi Koshiba[‡] Kenji Yasunaga[§]

December 30, 2021

Abstract

Secure Message Transmission (SMT) is a two-party cryptographic protocol by which the sender can securely and reliably transmit messages to the receiver using multiple channels. An adversary can corrupt a subset of the channels and commit eavesdropping and tampering attacks over the channels. In this work, we introduce a game-theoretic security model for SMT in which adversaries have some preferences for protocol execution. We define rational "timid" adversaries who prefer to violate security requirements but do not prefer the tampering to be detected.

First, we consider the basic setting where a single adversary attacks the protocol. We construct perfect SMT protocols against any rational adversary corrupting all but one of the channels. Since minority corruption is required in the traditional setting, our results demonstrate a way of circumventing the cryptographic impossibility results by a game-theoretic approach.

Next, we study the setting in which all the channels can be corrupted by multiple adversaries who do not cooperate. Since we cannot hope for any security if a single adversary corrupts all the channels or multiple adversaries cooperate maliciously, the scenario can arise from a game-theoretic model. We also study the scenario in which both malicious and rational adversaries exist.

1 Introduction

It is common to use the information network to send and receive messages. The physical channels between senders and receivers are composed of communication apparatuses, allowing adversaries to eavesdrop or tamper. While we usually use public-key cryptosystems to protect data over communication, their security needs computational assumptions. It is desirable to develop methods of achieving security in the information-theoretic sense.

In the two-party cryptographic setting, we typically assume a single communication channel between the parties. However, current network technologies can let many channels be available. Secure Message Transmission (SMT), introduced by Dolev et al. [12], is a cryptographic protocol for securely transmitting messages through multiple channels. Even if an adversary corrupts t out of n channels and commits eavesdropping and tampering over the corrupted channels, the messages are securely and correctly transmitted to the receiver by SMT. The requirements for SMT consist of *privacy* and *reliability*. The privacy guarantees that the adversary can obtain no information about the transmitted message, and the reliability guarantees that the receiver recovers the message sent

[*] This is the full version of [16] and [45].

[†] Saitama University, Japan.

[‡] Waseda University, Japan. Email: `tkoshiba@waseda.jp`

[§] Tokyo Institute of Technology, Japan. Email: `yasunaga@c.titech.ac.jp`

by the sender. If an SMT protocol satisfies both the requirements perfectly, the protocol is called a *perfect* SMT. Spini and Zémor [40] gave the most round-efficient perfect SMT. Dolev et al. [12] showed that any one-round perfect SMT must satisfy $t < n/3$ and that any perfect SMT whose round complexity is at least two must satisfy $t < n/2$. Garay and Ostrovsky [21] introduced the model of SMT *by Public Discussion* (SMT-PD), which allows transmission over an authentic and reliable public channel in addition to the n channels. Shi et al. [39] further studied SMT-PD and constructed a round-optimal perfect SMT-PD. In the context of *network coding*, similar but more general problems have been studied, and some schemes [43] can be seen as SMT protocols.

In the standard setting of cryptography, we assume the participants are either honest or malicious. The former will follow the protocol description honestly, and the latter may deviate from the protocol maliciously. In general, malicious behavior may be illegal and involve some risks, implying that adversaries in the standard cryptographic setting behave maliciously regardless of their risk. However, adversaries in real life may decide their behavior by taking the risk into account. To capture such situations, we incorporate game-theoretic *rational* participants into cryptography. Halpern and Teague [28] first studied the rational behavior of participants for secret sharing. Since then, rational secret sharing has been intensively studied [1, 22, 34, 35, 5, 15, 33]. Moreover, there have been many studies using game-theoretic analysis of cryptographic primitives/protocols, including two-party computation [4, 24], leader election [23, 2], Byzantine agreement [25], consensus [29], public-key encryption [44, 46], delegation of computation [7, 26, 9, 27, 10, 31], and protocol design [19, 20]. Among them, several works [25, 7, 26, 27, 19] used the rationality of adversaries to circumvent the impossibility results.

Groce et al. [25] studied the Byzantine agreement problem in the presence of a rational adversary. They showed that given some knowledge of the adversary's preference, a perfectly secure Byzantine agreement is possible for t corruptions among n players for any $t < n$. The security against $t \geq n/2$ corruptions is impossible in the standard adversary model.

This work shows that the impossibility results of SMT can also be circumvented by considering adversaries' rationality. As in the Byzantine agreement, we introduce a rational adversary for SMT who has some preference for the protocol execution outcome. More specifically, we define *timid* adversaries who prefer to violate the requirements of SMT but do not prefer the tampering to be detected. Such preferences can be justified if adversaries fear losing their corrupted channels when the protocol detects tampering.

1.1 Our Results

First, as the most basic setting, we define the security of perfect SMT in the presence of a *single* rational adversary. It is a natural extension of the standard cryptographic setting to the rational one. We show that the almost-reliable SMT-PD protocol of [39] works as a *perfect* SMT protocol. An intuitive reason is that timid adversaries do not have an incentive to attack the channels for fear of detection. Thus perfect reliability follows as well as perfect secrecy. To construct non-interactive SMT protocols, we consider *strictly* timid adversaries who prefer being undetected to violate the security requirements. We show that *robust* secret sharing schemes, which can detect forgery of shares with high probability, can be used as a non-interactive SMT protocol for such adversaries. Both protocols are perfectly secure against timid adversaries corrupting t out of n channels for any $t < n$, which is impossible in the standard setting of SMT protocols. We also present the impossibility of constructing SMT protocols against general timid adversaries corrupting $t \geq n/2$ channels. The result implies that the public discussion model is necessary for the first protocol, and

Table 1: Summary of the Results of Single-Adversary Setting

Adversary	**PD***	**Resiliency**	**Security**	**# Round**	**Construction**
Malicious	—	$t < n/3$	Perfect	1	Exist ([12])
Malicious	—	$t \geq n/3$	Perfect	1	Impossible ([12])
Malicious	—	$t < n/2$	Perfect	2	Exist ([36, 40])
Malicious	—	$t < n/2$	Almost Reliable	1	Exist ([43][1])
Timid	—	$t < n/2$	Perfect	1	Exist (Corollary 3)
Malicious	—	$t \geq n/2$	Perfect	2	Impossible ([12])
Malicious	✓	$t \geq n/2$	Almost Reliable	2	Impossible ([39])
Timid	—	$t \geq n/2$	Perfect	—	Impossible (Corollary 2)
Malicious	✓	$t < n$	Almost Reliable	3	Exist ([13, 21, 39, 18])
Timid	✓	$t < n$	Perfect	3	Exist (Theorem 4)
Strictly Timid	—	$t < n$	Perfect	1	Exist (Theorem 6)

*PD represents the public discussion model.

the strict timidness is necessary for the second protocol. The results are summarized in Table 1.

Next, we study the setting in which *multiple* timid adversaries may corrupt *all* the channels. More specifically, we assume that at least two adversaries exclusively corrupt subsets of the channels and prefer to violate the security requirements without being detected. We also assume that each adversary prefers other adversaries' tampering to be detected. This additional assumption makes rational adversaries avoid cooperating. If a single adversary corrupts all the channels, we cannot hope for any security of SMT. Thus, multiple conflicting adversaries are necessary for achieving security. We believe that the multiple-adversary setting is more realistic than the single-adversary one since it is difficult for the adversary to confirm that no other adversary exists. Also, protocols in the multiple setting need to equip the property to declare the tampering detection with channel identifiers. This required property is more desirable than a detection mechanism without channel identifiers, sufficient in the single setting. We show that secure SMT protocols exist even if such rational adversaries corrupt all the channels. The SMT-PD protocol of [39] also works in this setting as perfect SMT-PD. To construct perfect SMT protocols without public discussion, we employ the idea of *cheater-identifiable* secret sharing (CISS), in which every player who submits a forged share in the reconstruction phase can be identified. We construct a non-interactive SMT protocol based on the CISS of Hayashi and Koshiba [30]. Technically, our construction employs pairwise independent (a.k.a. strongly universal) hash functions instead of universal hash functions in [30]. Since the security requirements of CISS are not sufficient for proving the security of SMT against timid adversaries, 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 less than $n/2$ channels. We show that a variant of our CISS-based protocol works as a perfect SMT protocol for *strictly* timid adversaries, even if each adversary corrupts a majority of the channels.

Finally, we consider the setting in which a malicious adversary exists as well as rational adversaries. Namely, there are heterogeneous adversaries, all but one behave rationally, but one acts

[1] The paper studied more general problems of secure network coding. By considering a simple n-link network as in SMT and assuming the adversary who eavesdrops and tampers with the same links, the coding scheme of [43] gives a construction of an almost-reliable SMT protocol for $t < n/2$.

Table 2: Perfect SMT Protocols

Adversary	PD	Total Resiliency	# Adv.	Resiliency per Adv.	# Round	References
Malicious	—	$< n/3$	1	$< n/3$	1	[12]
Malicious	—	$< n/2$	1	$< n/2$	2	[36, 40]
Timid	—	$< n/2$	1	$< n/2$	1	Corollary 3
Timid	✓	$< n$	1	$< n$	3	Theorem 4
Strictly Timid	—	$< n$	1	$< n$	1	Theorem 6
Timid/Malicious	—	n	≥ 2	$< n/6$	1	Theorem 11
Timid	—	n	≥ 2	$< n/2$	1	Theorem 9
Timid	✓	n	≥ 2	$< n$	3	Theorem 8
Strictly Timid	—	n	≥ 2	$< n$	1	Theorem 10

maliciously. We believe this setting is preferable because the assumption that all adversaries are rational may not be realistic. 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 summarize constructions of perfect SMT protocols both for the single-adversary and the multiple-adversary settings in Table 2. The total resiliency is the maximum number of corrupted channels for which the protocol can achieve security.

We note that the single-adversary setting can be seen as a special case of the multiple-adversary setting. Some protocols for multiple adversaries may work against single adversaries. We show that Theorem 9 implies a non-interactive protocol for single adversaries that is secure against $t < n/2$ corruption (Corollary 3). This result circumvents the impossibility of constructing one-round protocols for $t \geq n/3$ in [12] by a game-theoretic consideration.

1.2 Related Work

The adversaries' behavior of avoiding detection has been used in the literature of multiparty computation. Franklin and Yung [14] defined the notion of *t-detectability*, which guarantees that no coalition of t parties can either learn any information about other $n - t$ parties' inputs or prevent the honest parties from detecting the tampering. Aumann and Lindell [6] introduced the notion of security against *covert adversaries*, who attempt to cheat but do not want to be caught with some prescribed probability. The underlying idea of covert adversaries is similar to that of timid adversaries in this work. However, there are several key differences. First, the goal of security notions in [14, 6] is to detect the adversary's tampering. These works do not consider what happens if tampering is detected. We provide a game-theoretic framework that guarantees perfect secrecy and reliability against adversaries trying to avoid detection. Second, adversaries in [14, 6] only try to learn private inputs of honest parties, while timid adversaries in this work try to violate both reliability and secrecy. In particular, since all protocols in this work achieve perfect secrecy, timid adversaries are essentially concerned about how to violate the reliability of the protocols. Such adversaries were not considered in [14, 6]. Also, as far as we know, security against covert adversaries can only be achieved against computationally bounded adversaries. We construct perfect SMT protocols against computationally unbounded adversaries.

Another efficiency metric discussed in the SMT literature is the *communication complexity* of the protocols [41, 3]. Minimizing the communication complexities of our protocols is an interesting future work.

1.3 Organization

Section 2 describe the definitions and the known results of secure message transmission. The security of SMT against a single adversary is given in Section 3, and our protocols are presented in Section 4. In Section 5, we show an impossibility result for general timid adversaries. We define the security of SMT against multiple adversaries in Section 6 and give the constructions of SMT protocols in Section 7. We study a mixed model of rational and malicious adversaries in Section 8. We conclude the paper in Section 9.

2 Secure Message Transmission

We assume that there are n channels between a sender $\mathcal{S}$ and a receiver $\mathcal{R}$. SMT protocols proceed in *rounds*. In each round, either $\mathcal{S}$ or $\mathcal{R}$ can send messages over the channels. The messages are delivered before the next round starts. The adversary $\mathcal{A}$ can corrupt at most t channels out of the n channels; such an adversary is referred to as *t-adversary*. On the corrupted channels, $\mathcal{A}$ can eavesdrop, block communication, or place any messages on them. We assume that $\mathcal{A}$ is *rushing*. Namely, $\mathcal{A}$ can decide the actions on the corrupted channels after observing the information sent on the corrupted channels. We consider computationally *unbounded* $\mathcal{A}$.

Let $\mathcal{M}$ be the message space. In SMT, $\mathcal{S}$ tries to transmit a message in $\mathcal{M}$ to $\mathcal{R}$, and $\mathcal{R}$ outputs the received 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 $\mathcal{R}$. An execution of Π can be completely characterized by the random coins of all the parties, namely, $\mathcal{S}$, $\mathcal{R}$, and $\mathcal{A}$, and the message M_S. Let $V_A(m, r_A)$ be the *view* of $\mathcal{A}$ when $M_S = m$, and $\mathcal{A}$ uses r_A as the random coins. Precisely, $V_A(m, r_A)$ consists of the messages sent over the corrupted channels when the protocol is run with $M_S = m$, and r_A, the random coins of $\mathcal{A}$.

We formally define the security requirements 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 change messages sent over the corrupted 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*

  $$\Delta(V_A(m_0, r_A), V_A(m_1, r_A)) \leq \varepsilon,$$

  where $\Delta(X, Y)$ denotes the statistical distance between two random variables X and Y over a finite set Ω, which is defined by

  $$\Delta(X, Y) = \frac{1}{2} \sum_{u \in \Omega} |\Pr[X = u] - \Pr[Y = u]|\,.$$

- 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 $\mathcal{A}$.

A protocol achieving $(0, 0)$-SMT is called *perfect*. If a protocol achieves $(0, \delta)$-SMT for small δ, it is called *almost-reliable* SMT.

Dolev et al. [12] characterized the trade-off between the achievability and the round complexity of perfect SMT.

Theorem 1 ([12])**.** *One-round perfect SMT protocols against t-adversary exist if and only if $t < n/2$. Also, multi-round perfect SMT protocols against t-adversary exist if and only if $t < n/3$.*

SMT by Public Discussion. In addition to the n channels, we may assume that $\mathcal{S}$ and $\mathcal{R}$ can use an authentic and reliable *public channel* on which messages are publicly accessible and guaranteed to be correctly delivered. Such protocols are referred to as SMT *by Public Discussion* (SMT-PD). Franklin and Wright [13] gave an impossibility result of SMT-PD by using different terminology. (See [17] for this fact.)

Theorem 2 ([13])**.** *Perfectly-reliable ($\delta = 0$) SMT-PD protocols against t-adversary exist only if $t < n/2$.*

Shi et al. [39] gave several impossibility results of SMT-PD and constructed a round-optimal SMT-PD protocol. We use their protocol. The description appears in Section 4.1.

3 SMT against a Single Rational Adversary

We define our security model of SMT protocol against a single rational adversary. The rationality of the adversary is characterized by a *utility function* that represents the preference of the adversary over possible outcomes of the protocol execution.

We can consider various preferences of the adversary regarding the SMT protocol execution. The adversary may prefer to violate the privacy or the reliability of SMT protocols. Also, the adversary may prefer to violate the above properties without being tampering detected. Here, we consider the adversary who prefers (1) to violate privacy, (2) to violate reliability, and (3) the tampering to be undetected.

To define the utility function, we specify the SMT game as follows.

The SMT Game. For an SMT protocol Π, we define our SMT game $\mathsf{Game}^{\mathsf{sngl}}(\Pi, \mathcal{A})$ against a single adversary $\mathcal{A}$. First, set three parameters $\mathsf{guess} = \mathsf{suc} = \mathsf{detect} = 0$. For 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, as in the usual SMT, the adversary $\mathcal{A}$ can corrupt at most t channels and tamper with any messages sent over the corrupted channels. If the sender or the receiver sends a special message "DETECT" during the execution, set $\mathsf{detect} = 1$. After running the protocol, the receiver outputs M_R, and the adversary outputs M_A. If $M_R = M_S$, set $\mathsf{suc} = 1$. If $M_A = M_S$, set $\mathsf{guess} = 1$. The outcome of the game is $(\mathsf{guess}, \mathsf{suc}, \mathsf{detect})$.

By following [15], we model the game where the adversary tries to guess the message chosen uniformly at random. In general, it is difficult to model the "real" game that the adversary attacks.

The above formulation can capture the situation that the adversary learns partial information of the message. If the partial information increases the probability of correctly guessing the message, the adversary obtains higher utility in the game.

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

Definition 2 (Utility)**.** *The utility $u(\mathcal{A}, U)$ of the adversary $\mathcal{A}$ with utility function U is the expected value $\mathbb{E}[U(\mathsf{out})]$, where U is a function that maps the outcome $\mathsf{out} = (\mathsf{guess}, \mathsf{suc}, \mathsf{detect})$ of the game $\mathsf{Game}^{\mathsf{sngl}}(\Pi, \mathcal{A})$ to real values and the probability is taken over the randomness of the game.*

The utility function U characterizes the type of adversaries. If the adversary has the preferences (1)-(3) as above, the utility function may have the property such that for any two outcomes $\mathsf{out} = (\mathsf{guess}, \mathsf{suc}, \mathsf{detect})$ and $\mathsf{out}' = (\mathsf{guess}', \mathsf{suc}', \mathsf{detect}')$ of the SMT game,

1. $U(\mathsf{out}) > U(\mathsf{out}')$ if $\mathsf{guess} > \mathsf{guess}'$, $\mathsf{suc} = \mathsf{suc}'$, and $\mathsf{detect} = \mathsf{detect}'$;

2. $U(\mathsf{out}) > U(\mathsf{out}')$ if $\mathsf{guess} = \mathsf{guess}'$, $\mathsf{suc} < \mathsf{suc}'$, and $\mathsf{detect} = \mathsf{detect}'$;

3. $U(\mathsf{out}) > U(\mathsf{out}')$ if $\mathsf{guess} = \mathsf{guess}'$, $\mathsf{suc} = \mathsf{suc}'$, and $\mathsf{detect} < \mathsf{detect}'$.

Based on the utility function of the adversary, we define the security of SMT against rational adversaries. In particular, regarding the security requirements, we only consider perfect SMT.

Definition 3 (PSMT against a Rational Adversary)**.** *An SMT protocol* Π *is* perfectly secure against a rational t-adversary with utility function U *if there is a t-adversary $\mathcal{B}$ such that*

1. Perfect security*: Π is $(0, 0)$-SMT against $\mathcal{B}$; and*

2. Nash equilibrium*: $u(\mathcal{A}, U) \leq u(\mathcal{B}, U)$ for any t-adversary $\mathcal{A}$.*

The perfect security guarantees that an adversary $\mathcal{B}$ is *harmless*. The Nash equilibrium guarantees that no adversary $\mathcal{A}$ can gain more utility than $\mathcal{B}$. Thus, the above security implies that no adversary $\mathcal{A}$ can gain more utility than the harmless adversary. Namely, the adversary does not have an incentive to deviate from the strategy of the harmless adversary $\mathcal{B}$.

In the security proof of our protocol, we will consider an adversary $\mathcal{B}$ who does not tamper with any messages on the channels and outputs a random message from $\mathcal{M}$ as M_A. We call such $\mathcal{B}$ a *random guessing* adversary. If $\mathcal{B}$ is random guessing, then the perfect security immediately follows from the correctness property of Π.

Timid Adversaries

We construct secure protocols against *timid* adversaries, who prefer to violate the security requirements of SMT protocols and do not prefer the tampering to be detected. More formally, the utility function of such adversaries should have properties such that

1. $U(\mathsf{out}) > U(\mathsf{out}')$ if $\mathsf{guess} = \mathsf{guess}'$, $\mathsf{suc} < \mathsf{suc}'$, and $\mathsf{detect} = \mathsf{detect}'$; and

2. $U(\mathsf{out}) > U(\mathsf{out}')$ if $\mathsf{guess} = \mathsf{guess}'$, $\mathsf{suc} = \mathsf{suc}'$, and $\mathsf{detect} < \mathsf{detect}'$,

where $\mathsf{out} = (\mathsf{guess}, \mathsf{suc}, \mathsf{detect}, \mathsf{abort})$ and $\mathsf{out}' = (\mathsf{guess}', \mathsf{suc}', \mathsf{detect}', \mathsf{abort}')$ are the outcomes of the SMT game. Let U_{timid} be the set of utility functions that satisfy the above conditions.

Also, timid adversaries may prefer being undetected to violating security. Such adversaries have the following utility:

3. $U(\mathsf{out}) > U(\mathsf{out}')$ if $\mathsf{guess} = \mathsf{guess}'$, $\mathsf{suc} > \mathsf{suc}'$, and $\mathsf{detect} < \mathsf{detect}'$.

Let $U_{\mathsf{st\text{-}timid}}$ be the set of utility functions satisfying the above three conditions. An adversary is *timid* if his utility function is in U_{timid}, and *strictly timid* if the utility function is in $U_{\mathsf{st\text{-}timid}}$.

In the analysis of our protocols, we need the following four values of utility:

- u_1 is the utility when $\Pr[\mathsf{guess} = 1] = \frac{1}{|\mathcal{M}|}$, $\mathsf{suc} = 0$, and $\mathsf{detect} = 0$;
- u_2 is the utility when $\Pr[\mathsf{guess} = 1] = \frac{1}{|\mathcal{M}|}$, $\mathsf{suc} = 1$, and $\mathsf{detect} = 0$;
- u_3 is the utility when $\Pr[\mathsf{guess} = 1] = \frac{1}{|\mathcal{M}|}$, $\mathsf{suc} = 0$, and $\mathsf{detect} = 1$;
- u_4 is the utility when $\Pr[\mathsf{guess} = 1] = \frac{1}{|\mathcal{M}|}$, $\mathsf{suc} = 1$, and $\mathsf{detect} = 1$.

It follows from the properties of utility functions in U_{timid} that $u_1 > \max\{u_2, u_3\}$ and $\min\{u_2, u_3\} > u_4$. For utility functions in $U_{\mathsf{st\text{-}timid}}$, it holds that $u_1 > u_2 > u_3 > u_4$.

The fact that all the utilities we use in our analysis are u_1, u_2, u_3, u_4 implies that all of our protocols achieve *perfect* privacy since the probability of guessing the message is $1/|\mathcal{M}|$ in every utility.

4 Protocols against a Timid Adversary

4.1 Protocol by Public Discussion

We show that an almost-reliable SMT-PD protocol proposed by Shi, Jiang, Safavi-Naini, and Tuhin [39] works as a perfect SMT-PD protocol against a timid adversary.

First, we describe the protocol and its proof overview.

The SJST Protocol. The protocol is based on the simple protocol for *static* adversaries where the sender sends a random key R_i over the ith channel for each $i \in \{1, \dots, n\}$, and the encrypted message $c = m \oplus R_1 \oplus \cdots \oplus R_n$ over the public channel. Suppose that the adversary $\mathcal{A}$ sees the messages sent over the corrupted channels and does not change them. Since $\mathcal{A}$ 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 $\mathcal{A}$. Thus, the message m can be securely encrypted and reliably sent through the public channel. The SJST protocol employs a mechanism for detecting the adversary's tampering by using hash functions to cope with *active* adversaries, who may change messages sent over the corrupted channels. Specifically, the *pairwise independent* hash functions (see Appendix A) 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 ith 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. 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 give a formal description of the SJST protocol in Figure 1, a three-round protocol that achieves reliability with $\delta = (n-1) \cdot 2^{1-\ell}$, where ℓ is the length of hash values.

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 ith channel.
2. For each $i \in \{1, \dots, n\}$, $\mathcal{R}$ receives (r'_i, R'_i) through the ith 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 $m = c \oplus (\bigoplus_{v_i=0} R_i)$.

Figure 1: The SJST Protocol

Theorem 3 ([39]). *The SJST protocol is $(0, (n-1) \cdot 2^{1-\ell})$-SMT against t-adversary for any $t < n$.*

One can find a complete proof of the above theorem in [39]. 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 ith 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. 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 in the second and the third rounds, the adversary can tamper with channels only in 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. The property of pairwise independent hash functions (Appendix A) implies that the above event happens with probability at most $(n-1)2^{1-\ell}$. Thus, the protocol achieves reliability with $\delta = (n-1) \cdot 2^{1-\ell}$.

For our purpose, we slightly modify the SJST protocol such that in the second and the third rounds, if $b_i = 1$ in B or $v_j = 1$ in V for some $i, j \in \{1, \dots, n\}$, the special message "DETECT" is also sent. We clarify the parameters of the SJST protocol to work as SMT against timid adversaries.

Theorem 4. *If the parameter ℓ in the SJST protocol satisfies*

$$\ell \geq \max\left\{1 + \log_2 t + \log_2 \frac{u_3 - u_4}{u_2 - u_4 - \alpha}, 1 + \frac{1}{t}\log_2 \frac{u_1 - u_3}{\alpha}\right\}$$

for some $\alpha \in (0, u_2 - u_4)$, then the protocol is perfectly secure against a rational t-adversary with utility function $U \in U_{\mathsf{timid}}$ for any $t < n$.

Proof. The perfect security of Definition 3 immediately follows by letting $\mathcal{B}$ be a random guessing adversary. We show that the strategy of $\mathcal{B}$ is a Nash equilibrium. Note that $u(\mathcal{B}, U) = u_2$, since $\Pr[\mathsf{guess} = 1] = \Pr[M_A = M_S] = 1/|\mathcal{M}|$ in the SMT game. Thus, it is sufficient to show that $u(\mathcal{A}, U) \leq u_2$ for any t-adversary $\mathcal{A}$. Also, note that, since the SJST protocol achieves the perfect privacy, it holds that $\Pr[\mathsf{guess} = 1] = 1/|\mathcal{M}|$ for any t-adversary.

Messages in the second and the third rounds are sent through the public channel. Thus, $\mathcal{A}$ can tamper with messages only in the first round. If $\mathcal{A}$ changes the lengths of r_i and R_i, the tampering of the ith channel will be detected. Such channels are simply ignored in the second and third rounds. Thus, such tampering cannot increase the utility. Hence, we assume that $\mathcal{A}$ does not change the lengths of r_i and R_i in the first round.

Suppose that $\mathcal{A}$ corrupts some t channels in the first round. Namely, there are exactly t distinct i's such that $(r'_i, R'_i) \neq (r_i, R_i)$. Note that the tampering on the ith channel such that $r'_i \neq r_i$ and $R'_i = R_i$ does not increase the probability that $\mathsf{suc} = 0$, but may increase the probability of detection. Thus, we also assume that $R'_i \neq R_i$ for all the corrupted channels. We define the following three events:

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

Note that all the events are disjoint, and either event should occur. Namely, we have that $\Pr[E_1] + \Pr[E_2] + \Pr[E_3] = 1$. It follows from the discussion in Appendix A 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, we have that $\Pr[E_1] = 2^{(1-\ell)t}$. Similarly, we obtain that $\Pr[E_3] = (1 - 2^{1-\ell})^t$. Note that the utility when E_1 occurs is at most u_1. Also, the utilities when E_2 and E_3 occur are at most u_3 and u_4, respectively. Therefore, the utility of $\mathcal{A}$ satisfies

$$\begin{aligned}
u(\mathcal{A}, U) &\leq u_1 \cdot \Pr[E_1] + u_3 \cdot \Pr[E_2] + u_4 \cdot \Pr[E_3] \\
&= u_3 + (u_1 - u_3) \Pr[E_1] - (u_3 - u_4) \Pr[E_3] \\
&\leq u_3 + (u_1 - u_3) 2^{(1-\ell)t} - (u_3 - u_4) \left(1 - t2^{1-\ell}\right) \\
&\leq u_3 + \alpha - (u_3 - u_4) \left(1 - t2^{1-\ell}\right) && (1) \\
&\leq u_2, && (2)
\end{aligned}$$

where we use the relations $\ell \geq 1 + \frac{1}{t} \log_2 \frac{u_1 - u_3}{\alpha}$ and $\ell \geq 1 + \log_2 t + \log_2 \frac{u_3 - u_4}{u_2 - u_4 - \alpha}$ in (1) and (2), respectively. The utility of $\mathcal{A}$ is at most u_2, and hence the statement follows. □

If $u_2 > u_3$, which holds for strictly timid adversaries, by choosing $\alpha = u_2 - u_3$, the condition on ℓ is that

$$\ell \geq \max\left\{1 + \log_2 t, 1 + \frac{1}{t} \log_2 \frac{u_1 - u_3}{u_2 - u_3}\right\}.$$

4.2 Protocol against a Strictly Timid Adversary

We show that, under the condition that $u_2 > u_3$, a *robust secret sharing scheme* gives a non-interactive perfect SMT protocol. Namely, we can construct a non-interactive protocol for strictly timid adversaries.

4.2.1 Robust Secret Sharing

Secret sharing, introduced by Shamir [38] and Blackley [8], enables us to distribute 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 4. *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 $|I| \leq t$,*

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

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

Shamir [38] 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 ith 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 f by polynomial interpolation, and outputs $f(0) = s$ as a recovered secret.

McEliece and Sarwate [37] 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 use the 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. This property is called *robustness*. Although robustness is a desirable property, it is known that robust secret sharing is impossible when $t/2$ shares are tampered with [32].

In this work, we need a weaker notion of robustness in which any tampering actions should be detected with high probability. Such robust secret sharing was studied by Cramer et al. [11]. They introduced the notion of *algebraic manipulation detection (AMD) codes* and presented a simple way for constructing robust secret sharing from *linear* secret sharing and AMD codes. The robustness required for our protocol is slightly different from the one defined in [11][2].

Definition 5. *Let t, n be positive integers with $t < n$. A (t, n, δ)-*robust 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 $|I| > t$,*

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

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

[2] The robustness in [11] requires that the output of the reconstruction algorithm should be either the original message or the failure symbol with high probability. Namely, it is allowed to recover the original message even if some shares are tampered with. In Definition 5, we require that if some shares are tampered with, the output of the reconstruction algorithm should be the failure symbol.

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

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

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

- Robustness*: For any $s \in \mathcal{G}$ and $I \subseteq \{1, \ldots, n\}$ with $|I| \leq t$ and adversary $\mathcal{A}$, if $\tilde{s}_i \neq s_i$ for some $i \in \{1, \ldots, n\}$,*

  $$\Pr\left[\mathsf{Reconst}\left(\{i, \tilde{s}_i\}_{i \in \{1,\ldots,n\}}\right) \neq \perp\right] \leq \delta,$$

  where

  $$\tilde{s}_i = \begin{cases} \mathcal{A}(i, s, \{s_i\}_{i \in I}) & \text{if } i \in I \\ s_i & \text{if } i \notin I \end{cases}$$

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

We can see that the construction of [11] satisfies the above definition. Specifically, we have the following theorem, which will be used in our protocol in Section 4.2.2. See Appendix B for the proof.

Theorem 5. *Let $\mathbb{F}$ be a finite field of size q and characteristic p, and d an integer such that $d+2$ is not divisible by p. For any positive integers t and n satisfying $t < n \leq qd$, there is an explicit and efficient scheme of $(t, n, (d+1)/q)$-robust secret sharing with range $\mathbb{F}^d$, where each share is an element of $\mathbb{F}^{d+2}$.*

4.2.2 Our Protocol

Let $(\mathsf{Share}, \mathsf{Reconst})$ be a (t, n, δ)-robust secret sharing scheme with range $\mathcal{M}$. In the protocol, given a message $m \in \mathcal{M}$, the sender generates n shares $(s_1, \ldots, s_n)$ by $\mathsf{Share}(m)$, and sends each s_i over the ith channel. The receiver simply recovers the message by $\mathsf{Reconst}(\{i, \tilde{s}_i\}_{i \in \{1,\ldots,n\}})$, where $\tilde{s}_i$ is the received message over the ith channel.

Theorem 6. *The above protocol using a (t, n, δ)-robust secret sharing scheme is perfectly secure against a rational t-adversary with utility function $U \in U_{\mathsf{st\text{-}timid}}$ if U satisfies $u_2 > u_3$ and*

$$\delta \leq \frac{u_2 - u_3}{u_1 - u_3}.$$

Proof. As in the proof of Theorem 4, we consider a random guessing adversary $\mathcal{B}$. Then, the perfect security immediately follows.

We show that for any t-adversary $\mathcal{A}$, $u(\mathcal{A}, U) \leq u(\mathcal{B}, U)$. As discussed in the proof of Theorem 4, it is sufficient to prove that $u(\mathcal{A}, U) \leq u_2$ for any $\mathcal{A}$. Since the underlying secret sharing has the perfect privacy, we have that $\Pr[\mathsf{guess} = 1] = 1/|\mathcal{M}|$ for any t-adversary. Suppose $\mathcal{A}$ corrupts some t channels and alters some messages s_i into different $\tilde{s}_i$. It follows from the robustness of secret sharing that the tampering is detected with probability at least $1 - \delta$, in which case the secret is not recovered. Thus, the utility of $\mathcal{A}$ is

$$u(\mathcal{A}, U) \leq (1 - \delta)u_3 + \delta u_1 \leq u_2, \tag{3}$$

where (3) follows from the assumption. Therefore, the statement follows. □

The following corollary immediately follows.

Corollary 1. *Let $\mathbb{F}$ be a finite field of size $q = 2^\ell$, and d be any odd integer. The non-interactive protocol based on Theorem 5 is an SMT protocol with message space $\mathbb{F}^d$ that is perfectly secure against a rational t-adversary with utility function $U \in U_{\mathsf{st\text{-}timid}}$ for any $t < n \leq 2^\ell d$ if*

$$\ell \geq \log_2(d+1) + \log_2 \frac{u_1 - u_3}{u_2 - u_3}.$$

5 Impossibility Result for General Timid Adversaries

We show that no SMT protocol is secure against a general timid t-adversary for $t \geq n/2$ without the public channel. The result implies that using the public channel in Theorem 4 is necessary for achieving $t \geq n/2$. It also demonstrates the necessity of restricting the utility in Theorem 6 for constructing protocols for $t \geq n/2$ without using the public channel.

Theorem 7. *For any SMT protocol without using the public channel that is perfectly secure against a rational t-adversary with utility function $U \in U_{\mathsf{timid}}$, if U has the relation*

$$u_2 < \frac{1}{2}\left(1 - \frac{1}{|\mathcal{M}|}\right) u_3$$

then $t < n/2$, where $\mathcal{M}$ is the message space of the protocol.

Proof. Let Π be a protocol in the statement. We construct a t-adversary $\mathcal{A}$ for $t = \lceil n/2 \rceil$ that can successfully attack Π. For simplicity, we assume that $n = 2t$.

Let $\mathcal{B}$ be a random guessing adversary. Since Π is $(0,0)$-SMT against $\mathcal{B}$, it holds that $u(\mathcal{B}, U) \leq u_2$. We show the existence of a t-adversary $\mathcal{A}$ that achieves $u(\mathcal{A}, U) > u_2$, which implies that Π cannot achieve a Nash equilibrium.

In the SMT game, a message $m \in \mathcal{M}$ is randomly chosen, and, on input m, Π generates $(s_1^j, \ldots, s_n^j)$ for $j = 1, \ldots,$ where s_i^j is the message to be sent over the ith channel in the jth round. In the game, $\mathcal{A}$ does the following:

- Randomly choose $I \subseteq \{1, \ldots, n\}$ such that $|I| = t$, and corrupt the ith channel for every $i \in I$.
- Randomly choose $\tilde{m} \in \mathcal{M}$, and simulate Π on input $\tilde{m}$. Let $\tilde{s}_i^j$ be the message generated for the ith channel in the jth round.
- In each round j, for every $i \in I$, on receiving s_i^j through the ith channel, exchange s_i^j for $\tilde{s}_i^j$.

For this attack, the receiver cannot distinguish which message, m or $\tilde{m}$, was originally transmitted by the sender since both messages for m and $\tilde{m}$ are equally mixed. Hence, the probability that $\mathsf{suc} = 1$, denoted by p_s, is at most

$$p_s \leq \frac{1}{2}\left(1 - \frac{1}{|\mathcal{M}|}\right) + \frac{1}{|\mathcal{M}|} = \frac{1}{2}\left(1 + \frac{1}{|\mathcal{M}|}\right),$$

where $1/|\mathcal{M}|$ comes from the event that $\tilde{m} = m$.

Let p_d be the probability that Π outputs "DETECT" messages during the execution against the above attack. Without loss of generality, we assume that if Π does not output "DETECT" messages, the receiver outputs some message at the end of the protocol. If the tampering of $\mathcal{A}$ is not detected, the utility of $\mathcal{A}$ is at least u_1 with probability $1 - p_s$, and at least u_2 with probability p_s. If some tampering is detected, there can be two cases: (1) the receiver does not output any message, and (2) the receiver outputs some message. In case (1), the utility of $\mathcal{A}$ is u_3. In case (2), the probability that the $\mathsf{suc} = 1$ is at most p_s by the same argument as above. Hence, the utility of $\mathcal{A}$ when the tampering was detected is at least $(1 - p_s)u_3$. Thus, the utility of $\mathcal{A}$ in the SMT game is at least

$$\begin{aligned} u(\mathcal{A}, U) &\geq (1 - p_d)\left((1 - p_s)u_1 + p_s u_2\right) + p_d(1 - p_s)u_3 \\ &= (1 - p_s)u_1 + p_s u_2 - p_d\left((1 - p_s)u_1 + p_s u_2 - (1 - p_s)u_3\right) \\ &\geq (1 - p_s)u_3 \qquad (4) \\ &\geq \frac{1}{2}\left(1 - \frac{1}{|\mathcal{M}|}\right)u_3 \\ &> u_2, \qquad (5) \end{aligned}$$

where (4) follows from the fact that $p_d \leq 1$ and $(1-p_s)u_1+p_s u_2-(1-p_s)u_3 \geq 0$, and the assumption on U is used in (5). Therefore, Π does not satisfy the PSMT security for $t \geq n/2$.

When $n = 2t - 1$, the same attack as the above $\mathcal{A}$ can be implemented by invalidating the nth channel by substituting $\perp$ for every message over the nth channel. □

The theorem gives the following corollary.

Corollary 2. *There is no SMT protocol without a public channel that is perfectly secure against a rational t-adversary with utility function U for every $U \in U_{\mathsf{timid}}$ and $t \geq \lceil n/2 \rceil$.*

6 SMT against Multiple Rational Adversaries

We define our security model of SMT in the presence of multiple rational adversaries. For simplicity, we assume that each adversary corrupts different channels. A difference from the single-adversary model in Section 3 is that each adversary may prefer the tampering of other adversaries to be detected. The SMT game is slightly changed such that the protocol needs to declare the tampering detection with channel identifiers. With this functionality, adversaries will notice the detection of their tampering.

Suppose that there are λ adversaries $1, 2, \ldots, \lambda$ for $\lambda \geq 2$ and adversary $j \in \{1, \ldots, \lambda\}$ exclusively corrupts at most t_j channels out of the n channels for $t_j \geq 1$. We have $\sum_{j=1}^{\lambda} t_j \leq n$.

The SMT Game. For an SMT protocol Π, we define our SMT game $\mathsf{Game}^{\mathsf{mult}}(\Pi, \mathcal{A}_1, \ldots, \mathcal{A}_\lambda)$ against λ adversaries with the strategy profile $(\mathcal{A}_1, \ldots, \mathcal{A}_\lambda)$. First, set parameters $\mathsf{suc} = 0$ and $\mathsf{guess}_j = \mathsf{detect}_j = 0$ for every $j \in \{1, \ldots, \lambda\}$. For the message space $\mathcal{M}$ of Π, 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, the sender or the receiver may send a special message "DETECT at i" for $i \in \{1, \ldots, n\}$, meaning that some tampering was detected in channel i. Then, if adversary $j \in \{1, \ldots, \lambda\}$ 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 $\big(\mathsf{suc}, \{\mathsf{guess}_{j'}, \mathsf{detect}_{j'}\}_{j' \in \{1,\ldots,\lambda\}}\big)$.

Definition 6 (Utility). *The utility $U_j(\mathcal{A}_1, \dots, \mathcal{A}_\lambda, U)$ of adversary j when the strategy profile $(\mathcal{A}_1, \dots, \mathcal{A}_\lambda)$ and utility function U are employed is the expected value $\mathbb{E}[U(j, \mathsf{out})]$, where U is a function that maps index j and the outcome $\mathsf{out} = \big(\mathsf{suc}, \{\mathsf{guess}_{j'}, \mathsf{detect}_{j'}\}_{j' \in \{1,\dots,\lambda\}}\big)$ of the game $\mathsf{Game}^{\mathsf{mult}}(\Pi, \mathcal{A}_1, \dots, \mathcal{A}_\lambda)$ to real values and the probability is taken over the randomness of the game.*

We define the security of SMT protocols against multiple adversaries. For strategies $\mathcal{B}_1, \dots, \mathcal{B}_\lambda$, and $\mathcal{A}_j$, we denote by $(\mathcal{A}_j, \mathcal{B}_{-j})$ the strategy profile $(\mathcal{B}_1, \dots, \mathcal{B}_{j-1}, \mathcal{A}_j, \mathcal{B}_{j+1}, \dots, \mathcal{B}_\lambda)$.

Definition 7 (PSMT against Multiple Rational Adversaries). *An SMT protocol Π is* perfectly secure against rational $(t_1, \dots, t_\lambda)$-adversaries with utility function U *if there are t_j-adversary $\mathcal{B}_j$ for $j \in \{1, \dots, \lambda\}$ such that*

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

As in the analysis of protocols against a single adversary, we will consider a *random guessing* strategy profile $(\mathcal{B}_1, \dots, \mathcal{B}_\lambda)$ in which each $\mathcal{B}_j$ is a random guessing adversary. The perfect security for such a strategy profile immediately follows if the protocol satisfies the correctness.

Timid Adversaries

For the case of multiple adversaries, we define timid adversaries who prefer other adversaries' tampering to be detected. This property makes rational adversaries avoid cooperating with each other. Let $U^{\mathsf{mult}}_{\mathsf{timid}}$ be the set of utility functions that satisfy the following three 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 $k \in \{1, \dots, \lambda\} \setminus \{j\}$; 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 $j' \in \{1, \dots, \lambda\} \setminus \{k\}$,

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

Let $U^{\mathsf{mult}}_{\mathsf{st\text{-}timid}}$ be the set of utility functions satisfying the following condition in addition to the above three:

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 $k \in \{1, \dots, \lambda\} \setminus \{j\}$.

An adversary is said to be *timid* if his utility function is in $U^{\mathsf{mult}}_{\mathsf{timid}}$, and *strictly timid* if the utility function is in $U^{\mathsf{mult}}_{\mathsf{st\text{-}timid}}$. For $j \in \{1, \dots, n\}$ and $b \in \{0, 1\}$, we write $\mathsf{detect}_{-j} = b$ if $\mathsf{detect}_{j'} = b$ for every $j' \in \{1, \dots, n\} \setminus \{j\}$. In the analysis of our protocols, we use the following values of the utility of adversary $j \in \{1, \dots, \lambda\}$.

- 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 $\Pr[\mathsf{guess}_j = 1] = \frac{1}{|\mathcal{M}|}$, $\mathsf{suc} = 0$, $\mathsf{detect}_j = 1$, $\mathsf{detect}_{-j} = 1$;
- u'_3 is the utility when $\Pr[\mathsf{guess}_j = 1] = \frac{1}{|\mathcal{M}|}$, $\mathsf{suc} = 0$, $\mathsf{detect}_j = 1$, $\mathsf{detect}_{-j} = 0$;
- 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 $U^{\mathsf{mult}}_{\mathsf{timid}}$, it holds that $u'_1 > \max\{u'_2, u''_3\}$, $\min\{u'_2, u'_3\} > u'_4$, and $u''_3 > u'_3$. If the utility is in $U^{\mathsf{mult}}_{\mathsf{st\text{-}timid}}$, it holds that $u'_1 > u'_2 > u''_3 > u'_3 > u'_4$.

7 Protocols against Multiple Adversaries

7.1 Protocol by Public Discussion

We show that the SJST protocol of [39] gives a perfect SMT-PD protocol against multiple adversaries. As in Section 4.1, we 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 $i \in \{1, \dots, n\}$, the special message "DETECT at i" is also sent together.

Theorem 8. *For any $\lambda \geq 2$, let $t_1, \dots, t_\lambda$ be integers satisfying $t_1 + \cdots + t_\lambda \leq n$ and $1 \leq t_i \leq n-1$ for every $i \in \{1, \dots, \lambda\}$. If the parameter ℓ in the SJST protocol satisfies*

$$\ell \geq \max_{t \in \{t_1, \dots, t_\lambda\}} \left\{ 1 + \log_2 t + \log_2 \frac{u'_3 - u'_4}{u'_2 - u'_4 - \alpha}, 1 + \frac{1}{t} \log_2 \frac{u'_1 - u'_3}{\alpha} \right\}$$

for some $\alpha \in (0, u'_2 - u'_4)$, then the protocol is perfectly secure against rational $(t_1, \dots, t_\lambda)$-adversaries with utility function $U \in U^{\mathsf{mult}}_{\mathsf{timid}}$.

Proof. We note that the same argument as the proof of Theorem 4 can apply to the case of multiple adversaries. This is because even in the presence of multiple adversaries, as long as we consider a Nash equilibrium, each adversary j tries to maximize the utility by choosing a strategy $\mathcal{A}_j$ by assuming that all the other adversaries follow the random guessing strategy profile $\mathcal{B}_{-j}$. It is precisely the case analyzed in the proof of Theorem 4. Since the utility values u'_1, u'_2, u'_3, u'_4 corresponds to those of u_1, u_2, u_3, u_4, respectively, in Theorem 4, the statement follows. □

7.2 Protocol for Minority Corruptions

We provide a non-interactive SMT protocol based on secret-sharing and pairwise independent hash functions. See Section 4.2.1 and Appendix A for the definitions. The protocol is secure against multiple 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 Section 7.1.

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. 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 $j \neq i$, $h_i(s_j)$ is computed as an authentication tag for s_j. Then, $(s_i, h_i, \{h_i(s_j)\}_{j \neq i})$ will be sent through the ith channel. When s_i is modified to $s'_i \neq s_i$

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.

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 ith 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 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 $\tilde{m}$. Otherwise, output $\perp$.

Figure 2: Protocol 1 for Minority Corruption

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 $j \neq i$. Also, 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 $\{r_{j,i}\}_{j \neq i}$ for s_i will be sent through the ith channel so that only the ith channel reveals the information of s_i. Hence, the message sent through the ith 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 each adversary corrupts minorities of the channels, a single adversary cannot cause erroneous detection of silent adversaries. We give a formal description in Figure 2.

Theorem 9. *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 $U \in U_{\mathsf{timid}}^{\mathsf{mult}}$.

Proof. For $k \in \{1, \ldots, \lambda\}$, let $\mathcal{B}_k$ be a random guessing t_k-adversary. 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 ith 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) $\mathsf{detect}_k = 0$ 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 ith 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. □

Note that the single-adversary setting of Section 3 can be seen as a special case of the multiple-adversary setting. Namely, it is equivalent to the setting in which there are two adversaries, $\mathcal{A}_1$ and $\mathcal{A}_2$, such that $\mathcal{A}_1$ tries to violate the security requirements of SMT by corrupting at most $t \leq n-1$ channels, whereas $\mathcal{A}_2$, who corrupt $n - t \geq 1$ channels, does nothing for the protocol. Since Protocol 1 does not rely on the additional utility of u''_3 in the security analysis, it also gives an SMT protocol in the single-adversary setting.

Corollary 3. *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 a rational t-adversary with utility function $U \in U_{\mathsf{timid}}$ for any $t < n/2$.

7.3 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 corruption. In Protocol 1, the lists L_i 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 $n-1$, (2) the list L_i contains both i and j if the masked

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.

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

Figure 3: Protocol 2 for Majority Corruption

tag $h_j(s_i) \oplus r_{i,j}$ does not match $T_{i,j}$, and (3) the final list L of the corrupted channels is composed of the union of all the sets L_i, namely, $L = L_1 \cup \dots \cup L_n$. The threshold of $n-1$ can be achieved by Shamir's scheme. Intuitively, this protocol works for strictly timid adversaries because if some adversary tampers with messages over the ith channel, the tampering will be detected with high probability, and in that case, i must be included in the final list L. Since strictly timid adversaries prefer his tampering not to be detected, they will keep silent. We give a formal description of the protocol in Figure 3.

Theorem 10. *For any $\lambda \geq 2$, let $t_1, \dots, t_\lambda$ be integers satisfying $t_1 + \dots + t_\lambda \leq n$ and $1 \leq t_i \leq n-1$ for every $i \in \{1, \dots, \lambda\}$. If the parameter ℓ in Protocol 2 satisfies*

$$\ell \geq \log_2 \frac{u_1' - u_3''}{u_2' - u_3''} - 1,$$

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

Proof. For $k \in \{1, \dots, \lambda\}$, let $\mathcal{B}_k$ be a random guessing t_k-adversary. By the same reason as in the proof of Theorem 9, the protocol is perfectly secure against $(\mathcal{B}_1, \dots, \mathcal{B}_\lambda)$.

Next, we show that $(\mathcal{B}_1, \dots, \mathcal{B}_\lambda)$ is a Nash equilibrium. Let $\mathcal{A}_k$ be any t_k-adversary for $k \in \{1, \dots, \lambda\}$. As in the proof of Theorem 9, $\mathcal{A}_k$ needs to yield either (a) $\mathsf{suc} = 0$, or (b) $\mathsf{detect}_k = 0$ and $\mathsf{detect}_{k'} = 1$ for some $k' \neq k$. For the case of (a), $\mathcal{A}_k$ needs to corrupt the ith channel so that $\tilde{s}_i \neq s_i$. There is at least one index $i' \in \{1, \dots, n\}$ that is not corrupted by $\mathcal{A}_k$. Thus, by the property of pairwise independent hash functions, the index i is included in the list $L_{i'}$ with probability at least $1 - 2^{1-\ell}$, in which case the utility of $\mathcal{A}_k$ is at most u_3''. Hence, the expected utility is at most

$$U_k(\mathcal{A}_k, \mathcal{B}_{-k}) \leq 2^{-(\ell+1)} \cdot u_1' + \Big(1 - 2^{-(\ell+1)}\Big) \cdot u_3'',$$

which is at most u'_2 by assumption. For the case of (b), if some index i is in the final list L by the tampering by $\mathcal{A}_k$, then some channel $i' \neq i$ corrupted by $\mathcal{A}_k$ is also included in L. Thus, (b) cannot happen. Therefore, $(\mathcal{B}_1, \dots, \mathcal{B}_\lambda)$ is a Nash equilibrium. □

8 SMT against Malicious and Rational Adversaries

In the previous sections, we have discussed SMT against rational adversaries. We have assumed that all the adversaries behave rationally. The assumption may be strong because all of them can be characterized by the utility function we defined. This section discusses more realistic situations where adversaries may not behave rationally but maliciously.

8.1 Security Model

Without loss of generality, we assume that there are $\lambda \geq 2$ adversaries, and adversaries $1, \dots, \lambda - 1$ are rational, and adversary λ behaves maliciously. We use the same definitions of the SMT game and the utility function in Section 6. We define *robust* security against rational adversaries. A similar definition appeared in the context of rational secret sharing [1]. For strategies $\mathcal{B}_1, \dots, \mathcal{B}_{\lambda-1}, \mathcal{A}_\lambda, \mathcal{A}_j$ for $j \in \{1, \dots, \lambda - 1\}$, we denote by $(\mathcal{A}_j, \mathcal{B}_{-j}, \mathcal{A}_\lambda)$ the strategy profile $(\mathcal{B}_1, \dots, \mathcal{B}_{j-1}, \mathcal{A}_j, \mathcal{B}_{j+1}, \dots, \mathcal{B}_{\lambda-1}, \mathcal{A}_\lambda)$.

Definition 8 (Robust PSMT against Rational Adversaries)**.** *An SMT protocol* Π *is* t^*-robust perfectly secure against rational $(t_1, \dots, t_{\lambda-1})$-adversaries with utility function U *if there are* t_j-*adversary* $\mathcal{B}_j$ *for* $j \in \{1, \dots, \lambda - 1\}$ *such that for any* t_j*-adversary* $\mathcal{A}_j$ *for* $j \in \{1, \dots, \lambda - 1\}$ *and* t^**-adversary* $\mathcal{A}_\lambda$*,*

1. Perfect security*:* Π *is* $(0, 0)$*-SMT against* $(\mathcal{B}_1, \dots, \mathcal{B}_{\lambda-1}, \mathcal{A}_\lambda)$*, 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* $j \in \{1, \dots, \lambda - 1\}$ *in the SMT game.*

Compared to Definition 7, robust PSMT requires that the perfect security is achieved even in the presence of a malicious adversary $\mathcal{A}_\lambda$, and a strategy profile $(\mathcal{B}_1, \dots, \mathcal{B}_{\lambda-1}, \mathcal{A}_\lambda)$ is a Nash equilibrium for adversary $j \in \{1, \dots, \lambda - 1\}$.

8.2 Protocol against Malicious and Rational Adversaries

We show that a robust PSMT protocol can be constructed based on the protocol for minority corruption in Section 7.2. For t^*-robust against $(t_1, \dots, t_{\lambda-1})$-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 $j \in \{1, \dots, \lambda - 1\}$. 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. Thus, the majority voting can identify any tampering with high probability.

The formal description is given in Figure 4.

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 as long as at most $\lfloor (n-1)/3 \rfloor$ out of n shares are tampered. 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.

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 $m_i = \big(s_i, h_i, \{T_{i,j}\}_{j \in \{1,\ldots,n\} \setminus \{i\}}, \{r_{j,i}\}_{j \in \{1,\ldots,n\} \setminus \{i\}}\big)$ through the ith 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 $\mathsf{Reconst}(\{i, \tilde{s}_i\}_{i \in \{1,\ldots,n\}})$, send message "DETECT at i" for every $i \in L$, and output $\tilde{m}$. Otherwise, output $\perp$.

Figure 4: Protocol 3

For the security analysis, we define the values of the 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 11. *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 $U \in U^{\mathsf{mult}}_{\mathsf{timid}}$.

Proof. For $k \in \{1, \ldots, \lambda - 1\}$, let $\mathcal{B}_k$ be a random guessing adversary. 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 $\mathsf{suc} = 1$. To increase the utility of adversary k, $\mathcal{A}_k$ needs to get either (a) $\mathsf{suc} = 0$, or (b) $\mathsf{detect}_k = 0$, 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 ith 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 9, 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

$$U_k(\mathcal{A}_k, \mathcal{B}_{-k}, \mathcal{A}_\lambda) \leq (n+1)^2 \cdot 2^{-(\ell+1)} \cdot u_1^* + \left(1 - (n+1)^2 \cdot 2^{-(\ell+1)}\right) \cdot u_4^* \leq u_2^*,$$

where (u_1^*, u_2^*, u_4^*) is either (u_1', u_2', u_4') or (u_1'', u_2'', u_4''). The last inequality follows from the assumption.

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 jth 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. □

9 Conclusions

We have introduced game-theoretic security models in SMT and constructed perfect SMT protocols against rational timid adversaries. Several protocols could circumvent the known impossibility results in the traditional cryptographic model. We have also constructed perfect SMT protocols when multiple rational adversaries corrupt all the channels. The results have revealed that we may not need to guarantee that adversaries do not corrupt one resource/channel if they may not cooperate. A feature of our model is that the best strategy for adversaries is to behave harmlessly. Namely, adversaries rationally decide to do nothing for the protocols. Although this conclusion seems similar to the honest-but-curious adversary model, the difference is significant between the situations in which adversaries can potentially attack actively or not.

One of future work is to apply our game-theoretic models to other primitives and protocols. The model of timid adversaries might be useful for constructing more efficient and resilient protocols. It can be used to avoid the impossibility results in the traditional setting. Since rational adversaries in our models do not attack actively, it seems easier to construct protocols by composition. Another direction is to study the mixed model of malicious and rational adversaries. Since real-life situations may fall into this setting, it is beneficial to construct more efficient protocols than in the usual cryptographic setting.

Acknowledgments

This work was supported in part by JSPS Grants-in-Aid for Scientific Research Numbers 16H01705, 17H01695, 18K11159, 19K22849, and 21H04879. We thank Masahito Hayashi for discussions about the relations between SMT and secure network coding.

References

[1] I. Abraham, D. Dolev, R. Gonen, and J. Y. Halpern. Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In E. Ruppert and D. Malkhi, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006*, pages 53–62. ACM, 2006.

[2] I. Abraham, D. Dolev, and J. Y. Halpern. Distributed protocols for leader election: A game-theoretic perspective. *ACM Trans. Economics and Comput.*, 7(1):4:1–4:26, 2019.

[3] S. Agarwal, R. Cramer, and R. de Haan. Asymptotically optimal two-round perfectly secure message transmission. In C. Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 394–408. Springer, 2006.

[4] G. Asharov, R. Canetti, and C. Hazay. Toward a game theoretic view of secure computation. *J. Cryptology*, 29(4):879–926, 2016.

[5] G. Asharov and Y. Lindell. Utility dependence in correct and fair rational secret sharing. *J. Cryptology*, 24(1):157–202, 2011.

[6] Y. Aumann and Y. Lindell. Security against covert adversaries: Efficient protocols for realistic adversaries. *J. Cryptology*, 23(2):281–343, 2010.

[7] P. D. Azar and S. Micali. Super-efficient rational proofs. In M. J. Kearns, R. P. McAfee, and É. Tardos, editors, *Proceedings of the fourteenth ACM Conference on Electronic Commerce, EC 2013*, pages 29–30. ACM, 2013.

[8] G. R. Blakley. Safeguarding cryptographic keys. *Proc. of the National Computer Conference*, 48:313–317, 1979.

[9] M. Campanelli and R. Gennaro. Sequentially composable rational proofs. In M. H. R. Khouzani, E. A. Panaousis, and G. Theodorakopoulos, editors, *Decision and Game Theory for Security - 6th International Conference, GameSec 2015*, volume 9406 of *Lecture Notes in Computer Science*, pages 270–288. Springer, 2015.

[10] M. Campanelli and R. Gennaro. Efficient rational proofs for space bounded computations. In S. Rass, B. An, C. Kiekintveld, F. Fang, and S. Schauer, editors, *Decision and Game Theory for Security - 8th International Conference, GameSec 2017*, volume 10575 of *Lecture Notes in Computer Science*, pages 53–73. Springer, 2017.

[11] R. Cramer, Y. Dodis, S. Fehr, C. Padró, and D. Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In N. P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 471–488. Springer, 2008.

[12] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *J. ACM*, 40(1):17–47, 1993.

[13] M. K. Franklin and R. N. Wright. Secure communication in minimal connectivity models. *J. Cryptology*, 13(1):9–30, 2000.

[14] M. K. Franklin and M. Yung. Communication complexity of secure computation (extended abstract). In S. R. Kosaraju, M. Fellows, A. Wigderson, and J. A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, STOC '92*, pages 699–710. ACM, 1992.

[15] G. Fuchsbauer, J. Katz, and D. Naccache. Efficient rational secret sharing in standard communication networks. In D. Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010*, volume 5978 of *Lecture Notes in Computer Science*, pages 419–436. Springer, 2010.

[16] M. Fujita, K. Yasunaga, and T. Koshiba. Perfectly secure message transmission against rational timid adversaries. In L. Bushnell, R. Poovendran, and T. Basar, editors, *Decision and Game Theory for Security - 9th International Conference, GameSec 2018*, volume 11199 of *Lecture Notes in Computer Science*, pages 127–144. Springer, 2018.

[17] J. A. Garay, C. Givens, and R. Ostrovsky. Secure message transmission by public discussion: A brief survey. In Y. M. Chee, Z. Guo, S. Ling, F. Shao, Y. Tang, H. Wang, and C. Xing, editors, *Coding and Cryptology - Third International Workshop, IWCC 2011*, volume 6639 of *Lecture Notes in Computer Science*, pages 126–141. Springer, 2011.

[18] J. A. Garay, C. Givens, and R. Ostrovsky. Secure message transmission with small public discussion. *IEEE Trans. Inf. Theory*, 60(4):2373–2390, 2014.

[19] J. A. Garay, J. Katz, U. Maurer, B. Tackmann, and V. Zikas. Rational protocol design: Cryptography against incentive-driven adversaries. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013*, pages 648–657. IEEE Computer Society, 2013.

[20] J. A. Garay, J. Katz, B. Tackmann, and V. Zikas. How fair is your protocol?: A utility-based approach to protocol optimality. In C. Georgiou and P. G. Spirakis, editors, *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015*, pages 281–290. ACM, 2015.

[21] J. A. Garay and R. Ostrovsky. Almost-everywhere secure computation. In N. P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 4965 of *Lecture Notes in Computer Science*, pages 307–323. Springer, 2008.

[22] S. D. Gordon and J. Katz. Rational secret sharing, revisited. In R. D. Prisco and M. Yung, editors, *Security and Cryptography for Networks, 5th International Conference, SCN 2006*, volume 4116 of *Lecture Notes in Computer Science*, pages 229–241. Springer, 2006.

[23] R. Gradwohl. Rationality in the full-information model. In D. Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010*, volume 5978 of *Lecture Notes in Computer Science*, pages 401–418. Springer, 2010.

[24] A. Groce and J. Katz. Fair computation with rational players. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International*

Conference on the Theory and Applications of Cryptographic Techniques, volume 7237 of *Lecture Notes in Computer Science*, pages 81–98. Springer, 2012.

[25] A. Groce, J. Katz, A. Thiruvengadam, and V. Zikas. Byzantine agreement with a rational adversary. In A. Czumaj, K. Mehlhorn, A. M. Pitts, and R. Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012*, volume 7392 of *Lecture Notes in Computer Science*, pages 561–572. Springer, 2012.

[26] S. Guo, P. Hubácek, A. Rosen, and M. Vald. Rational arguments: single round delegation with sublinear verification. In M. Naor, editor, *Innovations in Theoretical Computer Science, ITCS'14*, pages 523–540. ACM, 2014.

[27] S. Guo, P. Hubácek, A. Rosen, and M. Vald. Rational sumchecks. In E. Kushilevitz and T. Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A*, volume 9563 of *Lecture Notes in Computer Science*, pages 319–351. Springer, 2016.

[28] J. Y. Halpern and V. Teague. Rational secret sharing and multiparty computation: extended abstract. In L. Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, STOC 2004*, pages 623–632. ACM, 2004.

[29] J. Y. Halpern and X. Vilaça. Rational consensus: Extended abstract. In G. Giakkoupis, editor, *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016*, pages 137–146. ACM, 2016.

[30] M. Hayashi and T. Koshiba. Universal construction of cheater-identifiable secret sharing against rushing cheaters based on message authentication. In *2018 IEEE International Symposium on Information Theory, ISIT 2018*, pages 2614–2618. IEEE, 2018.

[31] K. Inasawa and K. Yasunaga. Rational proofs against rational verifiers. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 100-A(11):2392–2397, 2017.

[32] Y. Ishai, R. Ostrovsky, and H. Seyalioglu. Identifying cheaters without an honest majority. In R. Cramer, editor, *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012*, volume 7194 of *Lecture Notes in Computer Science*, pages 21–38. Springer, 2012.

[33] A. Kawachi, Y. Okamoto, K. Tanaka, and K. Yasunaga. General constructions of rational secret sharing with expected constant-round reconstruction. *Comput. J.*, 60(5):711–728, 2017.

[34] G. Kol and M. Naor. Cryptography and game theory: Designing protocols for exchanging information. In R. Canetti, editor, *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008*, volume 4948 of *Lecture Notes in Computer Science*, pages 320–339. Springer, 2008.

[35] G. Kol and M. Naor. Games for exchanging information. In C. Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC 2008*, pages 423–432. ACM, 2008.

[36] K. Kurosawa and K. Suzuki. Truly efficient 2-round perfectly secure message transmission scheme. *IEEE Trans. Information Theory*, 55(11):5223–5232, 2009.

[37] R. J. McEliece and D. V. Sarwate. On sharing secrets and reed-solomon codes. *Commun. ACM*, 24(9):583–584, 1981.

[38] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[39] H. Shi, S. Jiang, R. Safavi-Naini, and M. A. Tuhin. On optimal secure message transmission by public discussion. *IEEE Trans. Information Theory*, 57(1):572–585, 2011.

[40] G. Spini and G. Zémor. Perfectly secure message transmission in two rounds. In M. Hirt and A. D. Smith, editors, *Theory of Cryptography - 14th International Conference, TCC 2016-B*, volume 9985 of *Lecture Notes in Computer Science*, pages 286–304, 2016.

[41] K. Srinathan, A. Narayanan, and C. P. Rangan. Optimal perfectly secure message transmission. In M. K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 545–561. Springer, 2004.

[42] M. N. Wegman and L. Carter. New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.

[43] H. Yao, D. Silva, S. Jaggi, and M. Langberg. Network codes resilient to jamming and eavesdropping. *IEEE/ACM Trans. Netw.*, 22(6):1978–1987, 2014.

[44] K. Yasunaga. Public-key encryption with lazy parties. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 99-A(2):590–600, 2016.

[45] K. Yasunaga and T. Koshiba. Perfectly secure message transmission against independent rational adversaries. In T. Alpcan, Y. Vorobeychik, J. S. Baras, and G. Dán, editors, *Decision and Game Theory for Security - 10th International Conference, GameSec 2019*, volume 11836 of *Lecture Notes in Computer Science*, pages 563–582. Springer, 2019.

[46] K. Yasunaga and K. Yuzawa. Repeated games for generating randomness in encryption. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 101-A(4):697–703, 2018.

A Pairwise Independent Hash Functions

Wegman and Carter [42] introduced the notion of pairwise independent (or strongly universal) hash functions and gave its construction.

Definition 9. *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 ([39])**.** *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 [42], 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

$$\Pr_{h \in H_{wc}}[c_1 \oplus h(x_1) = c_2 \oplus h(x_2)] = 2^{1-\ell} \tag{6}$$

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

B Proof of Theorem 5

To prove the theorem, we define the notion of *algebraic manipulation detection (AMD) codes* in which the security requirement is slightly different from that in [11] for our purpose.

Definition 10. *An (M, N, δ)-*algebraic manipulation detection (AMD) code *is a probabilistic function $E\colon \mathcal{S} \to \mathcal{G}$, where $\mathcal{S}$ is a set of size M and $\mathcal{G}$ is an additive group of order N, together with a decoding function $D\colon \mathcal{G} \to \mathcal{S} \cup \{\perp\}$ such that*

- Correctness*: For any $s \in \mathcal{S}$,* $\Pr[D(E(s)) = s] = 1$.
- Security*: For any $s \in \mathcal{S}$ and $\Delta \in \mathcal{G} \setminus \{0\}$,* $\Pr[D(E(s) + \Delta) \neq \perp] \leq \delta$.

An AMD code is called systematic *if $\mathcal{S}$ is a group, and the encoding is of the form*

$$E\colon \mathcal{S} \to \mathcal{S} \times \mathcal{G}_1 \times \mathcal{G}_2, s \mapsto (s, x, f(x, s))$$

for some function f and random $x \in \mathcal{G}_1$. The decoding function D of a systematic AMD code is given by $D(s', x', f') = s'$ if $f' = f(x', s')$, and $\perp$ otherwise.

Note that, for a systematic AMD code, the correctness immediately follows from the definition of the decoding function. The security requirement can be stated such that for any $s \in \mathcal{S}$ and $(\Delta_s, \Delta_x, \Delta_f) \in \mathcal{S} \times \mathcal{G}_1 \times \mathcal{G}_2 \setminus \{(0,0,0)\}$, $\Pr_x[f(s + \Delta_s, x + \Delta_x) = f(s, x) + \Delta_f] \leq \delta$.

We show that a systematic AMD code given in [11] satisfies the above definition.

Proposition 1. *Let $\mathbb{F}$ be a finite field of size q and characteristic p, and d any integer such that $d+2$ is not divisible by p. Define the encoding function $E\colon \mathbb{F}^d \to \mathbb{F}^d \times \mathbb{F} \times \mathbb{F}$ by $E(s) = (s, x, f(x, s))$ where*

$$f(x, s) = x^{d+2} + \sum_{i=1}^{d} s_i x^i$$

and $s = (s_1, \ldots, s_d)$. Then, the construction is a systematic $(q^d, q^{d+2}, (d+1)/q)$-AMD code.

Proof. We show that for any $s \in \mathbb{F}^d$ and $(\Delta_s, \Delta_x, \Delta_f) \in \mathbb{F}^d \times \mathbb{F} \times \mathbb{F} \backslash \{(0^d, 0, 0)\}$, $\Pr[f(s+\Delta_s, x+\Delta_x) = f(s,x) + \Delta_f] \leq \delta$. The event in the probability is that

$$(x + \Delta_x)^{d+2} + \sum_{i=1}^{d} s'_i (x + \Delta_x)^i = x^{d+2} + \sum_{i=1}^{d} s_i x^i + \Delta_f, \tag{7}$$

where s'_i is the ith element of $s + \Delta_s$. The left-hand side of (7) can be represented by

$$x^{d+2} + (d+2)\Delta_x x^{d+1} + \sum_{i=1}^{d} s'_i x^i + \Delta_x p(x)$$

for some polynomial $p(x)$ of degree at most d. Thus, (7) can be rewritten as

$$(d+2)\Delta_x x^{d+1} + \sum_{i=1}^{d} (s'_i - s_i) x^i + \Delta_x p(x) - \Delta_f = 0. \tag{8}$$

We discuss the probability that (8) happens when x is chosen uniformly at random. We consider the following cases:

1. When $\Delta_x \neq 0$, the coefficient of x^{d+1} is $(d+2)\Delta_x$, which is not zero by the assumption that $d+2$ is not divisible by p. Then, (8) has at most $d+1$ solutions x. Hence the event happens with probability at most $(d+1)/q$.

2. When $\Delta_x = 0$, we consider two subcases:

   (a) If $\Delta_s \neq 0$, then $s'_i - s_i \neq 0$ for some i. Hence (8) has at most d solutions x. Thus the event happens with probability at most d/p.

   (b) If $\Delta_s = 0$, (8) becomes $\Delta_f = 0$. Since $\Delta_f \neq 0$ for this case, the event cannot happen.

In every case, the event happens with probability at most $(d+1)/q$. Thus the statement follows. □

As discussed in [11], a robust secret sharing scheme can be obtained by combining an AMD code and a linear secret sharing scheme. Let $(\mathsf{Share}, \mathsf{Reconst})$ be a (t,n)-secret sharing scheme with range $\mathcal{G}$ that satisfies correctness and perfect privacy of Definition 5, where we drop the parameter δ for robustness. A *linear* secret sharing scheme has the property that for any $s \in \mathcal{G}$, $(s_1, \ldots, s_n) \in \mathsf{Share}(s)$, and vector $(s'_1, \ldots, s'_n)$, which may contain $\perp$ symbols, it holds that $\mathsf{Reconst}(\{i, s_i + s'_i\}_{i \in I}) = s + \mathsf{Reconst}(\{i, s'_i\}_{i \in I})$ for any $I \subseteq \{1, \ldots, n\}$ with $|I| > t$, where $\perp + x = x + \perp = \perp$ for all x. Examples of linear secret sharing schemes are Shamir's scheme [38] and the simple XOR-based $(n-1,n)$-scheme, in which secret $s \in \{0,1\}^n$ is shared by $(s_1, \ldots, s_n)$ for random $s_i \in \{0,1\}^n$ with the restriction that $s_1 \oplus \cdots \oplus s_n = s$.

We show that the same construction as in [11] works as a construction of robust secret sharing as per Definition 5.

Proposition 2. *Let* $(\mathsf{Share}, \mathsf{Reconst})$ *be a linear* (t,n)*-secret sharing scheme with range* $\mathcal{G}$ *that satisfies correctness and perfect privacy as per Definition 5, and let* (E, D) *be an* (M, N, δ)*-AMD code as per Definition 10 with* $|\mathcal{G}| = N$*. Then, the scheme* $(\mathsf{Share}', \mathsf{Reconst}')$ *defined by* $\mathsf{Share}'(s) = \mathsf{Share}(E(s))$ *and* $\mathsf{Reconst}'(S) = D(\mathsf{Reconst}(S))$ *is a* (t, n, δ)*-robust secret sharing scheme.*

Proof. Let $(s_1, \ldots, s_n) \in \mathsf{Share}'(s)$. Let $I \subseteq \{1, \ldots, n\}$ with $|I| \le t$, and $(\tilde{s}_1, \ldots \tilde{s}_n)$ be a sequence of shares satisfying the requirement for input shares in the robustness condition of Definition 5. We assume that $\tilde{s}_i = s_i + \Delta'_i$ for each $i \in \{1, \ldots, n\}$. Note that $\Delta'_i = 0$ for every $i \notin I$. Then,

$$\begin{aligned} \Pr\left[\mathsf{Reconst}'\left(\{i, \tilde{s}_i\}_{i\in\{1,\ldots,n\}}\right) \neq \perp\right] &= \Pr\left[D\left(E(s) + \mathsf{Reconst}(\{i, \Delta_i\}_{i\in\{1,\ldots,n\}})\right) \neq \perp\right] \\ &= \Pr\left[D\left(E(s) + \Delta\right) \neq \perp\right], \end{aligned}$$

where $\Delta = \mathsf{Reconst}\left(\{i, \Delta_i\}_{i\in\{1,\ldots,n\}}\right)$ is determined by the adversary. It follows from perfect privacy of the secret sharing scheme that Δ is independent of $E(s)$. Thus, if $\tilde{s}_i \neq s_i$ for some $i \in \{1, \ldots, n\}$, the probability is at most δ by the security of the AMD code. Hence, the statement follows. □

By combining Shamir's secret sharing scheme with range $\mathbb{F}^d$ and the AMD code of Proposition 1, the robust secret sharing scheme of Theorem 5 is obtained by Proposition 2.