New Results on 1-Round $(0, \delta)$-Secure Message Transmission against Generalized Adversary

No Author Given

No Institute Given

Abstract. In the Secure Message Transmission (SMT) problem, a sender $\mathcal{S}$ is connected to a receiver $\mathcal{R}$ through n disjoint paths in the network, a subset of which are controlled by an adversary with *unlimited computational power*. $\mathcal{S}$ wants to send a message m to $\mathcal{R}$ in a *private* and *reliable* way. Constructing secure and efficient SMT protocols has been extensively researched against a threshold adversary who can corrupt at most t out of n wires. SMT problem for a generalized adversary who can corrupt one out of many possible subsets is getting attention now-a-days due to its more practicality than a threshold adversary.
In this paper we focus on 1-round $(0, \delta)$ SMT protocols where privacy is perfect and the chance of protocol failure (receiver outputting **NULL**) is bounded by δ. These protocols are especially attractive because of their practical applications.
We first show an equivalence between secret sharing with cheating and a 1-round $(0, \delta)$-SMT against a generalized adversary satisfying certain conditions. This generalizes a similar equivalence against threshold adversaries. We use this equivalence to obtain a lower bound on the communication complexity of 1-round $(0, \delta)$-SMT against a generalized adversary under some conditions. We also derive a lower bound on the communication complexity of a general 1-round (0, 0)-SMT against a generalized adversary.
We finally give a construction using a linear secret sharing scheme and a special type of hash function. The protocol has almost optimal communication complexity and achieves this efficiency for a single message (does not require block of message to be sent).

1 Introduction

The **S**ecure **M**essage **T**ransmission (**SMT**) problem was introduced by Dolev, Dwork, Waarts and Yung in [9] to address the problem of secure communication between two nodes in an incomplete network. In the SMT problem, the sender $\mathcal{S}$ and the receiver $\mathcal{R}$ do not share a key but are connected by n 'wires', a subset of which are controlled by an adversary $\mathcal{A}$ with unlimited computational power. Wires are abstractions of node-disjoint paths between $\mathcal{S}$ and $\mathcal{R}$. The sender $\mathcal{S}$ wants to send a message m to a receiver $\mathcal{R}$ in a 'private' and 'reliable' way. 'Private' means that the adversary should not learn any information about m and 'reliable' means that $\mathcal{R}$ will receive the same message m that $\mathcal{S}$ has sent. Security of an SMT protocol means achieving both privacy and reliability.

A *perfectly secure message transmission (PSMT)* guarantees that $\mathcal{R}$ always receives the sent message and the adversary never learns anything about the message.

One of the main motivations of studying SMT has been to reduce connectivity requirements in secure multi-party protocols in unconditional setting [2, 3, 20]. These protocols assume reliable and secure channels between every two nodes. This assumption cannot be satisfied in many real life scenarios and so, SMT is used to simulate a secure channel between nodes using redundant paths in the network. Algorithms and techniques developed in the study of SMT, and in particular one round protocols, have found other applications including in key distribution and in particular strengthening keys shared between nodes in sensor networks [22, 23].

Franklin and Wright relaxed the original definition of PSMT and proposed (ε, δ)-SMT $(0 \leq \varepsilon, \delta \leq 1)$ [12] where privacy and reliability losses are bounded by ϵ and δ respectively. Relaxing security and reliability reduces connectivity requirement and results in more efficient protocols.

Motivation of this Work.

Secure message transmission problem against a threshold adversary has been extensively researched in the literature. [9, 12, 14]. General adversary provides a flexible way of modeling real life adversaries and so it is important to develop the SMT theory to the case of generalized adversaries. This is the main motivation of this work.

Our Results. We present a number of results.

- We first show an equivalence between secret sharing with cheating and a 1-round (0, δ)-SMT against generalized adversary, assuming SMT and the adevrsary structure satisfy certain conditions. In particular the decoding furnction of SMT must be of a special form which is called canonical. This result generalizes a similar result for threshold adversaries due to Kurosawa and Suzuki [14]. Using this equivalence we will derive a lower bound on the communication complexity of a canonical 1-round (0, δ)-SMT against a generalized adversary.
- We derive a lower bound on the communication complexity of a 1-round (0, 0)-SMT against a generalized adversary. The bound is the first of this kind for generalized adversary case. A similar bound on the communication complexity was known for the case of threshold adversary.
- We finally give the construction of a 1-round (0, δ)-SMT protocol with security against a generalized adversary. The protocol is simpler than the other known protocol with the same property [8]. In particular it can be used for transmission of a single message, while in the case of the protocol of [8], efficiency is guaranteed only when a block of messages is sent. Both protocols are efficient (polynomial in the size of the adversary structure).

Organization of the paper.

We recall the basic definitions of secure message transmission, secret sharing, secret sharing with cheating, and linear secret sharing in Section 2. In Section

3 we show the equivalence between 1-round $(0, \delta)$-SMT and secret sharing with cheating against a generalized adversary under some assumptions. In Section 4 we derive the lower bound on the communication complexity of 1-round $(0, \delta)$-SMT from the shown equivalence. We also showed the lower bound on the communication complexity of 1-round $(0, 0)$-SMT in Section 4. Finally in Section 5, we designed a 1-round $(0, \delta)$-SMT protocol using a secret sharing with cheating scheme.

2 Preliminaries

Communication Model. We consider a *synchronous*, *incomplete* network. The sender $\mathcal{S}$ and the receiver $\mathcal{R}$ are connected by n vertex-disjoint paths, also known as wires or channels. Both $\mathcal{S}$ and $\mathcal{R}$ are honest. The goal is for $\mathcal{S}$ to send a message m, drawn from the message space $\mathcal{M}$, to $\mathcal{R}$ such that $\mathcal{R}$ receives it correctly and privately.

The network is undirected and wires are two-way. SMT protocols proceeds in one or more rounds. In a round, a message is sent by either $\mathcal{S}$ or $\mathcal{R}$ to the other party over the wires. Messages are received by the recipient of the round before the next round starts. We consider only 1-round in this work, where the sender sends the message to the receiver.

Adversary Model. We consider an adversary $\mathcal{A}$ having *unlimited* computational power who can corrupt a subset of nodes in the network. Honest nodes forward the received messages to the next nodes on the path. A path (wire) that includes a corrupted node is controlled by the adversary. Corrupted nodes can fully control the corrupted wires and arbitrarily eavesdrop, modify or block messages sent over them. $\mathcal{A}$ is *adaptive* and can corrupt wires any time during the protocol execution and after observing communications over the wires that she has corrupted so far. $\mathcal{A}$ is also *rushing*, i.e., in each round it sees the messages sent by $\mathcal{S}$ and $\mathcal{R}$ over the corrupted wires before deciding on the messages to be sent over those wires in that round. $\mathcal{S}$ and $\mathcal{R}$ *do not know which wires are corrupted.*

Notation. $\mathcal{M}$ be the message space from which messages are chosen according to a probability distribution $\Pr(m)$. Let $m_{\mathcal{S}}$ be the message randomly selected by $\mathcal{S}$. We assume $\mathcal{M}$ and $\Pr(m)$ are known in advance to all parties including the adversary. Let $R_{\mathcal{A}}$ be the random coins used by $\mathcal{A}$ to choose one set of wires in the adversary structure Γ to corrupt.

In an execution of an SMT protocol Π, $\mathcal{S}$ draws $M_{\mathcal{S}}$ from $\mathcal{M}$ using the distribution $\Pr(m)$, and aims to send it to $\mathcal{R}$ privately and reliably. We assume that by the end of the protocol, $\mathcal{R}$ outputs a message $M_{\mathcal{R}} \in \mathcal{M}$ or **NULL** and so, an execution is completely determined by the random coins selected by all parties and the messages selected by the sender.

2.1 Secure Message Transmission

Let the sender $\mathcal{S}$ and the receiver $\mathcal{R}$ are connected by a set of n wires $\mathcal{W} = \{w_1, w_2, ..., w_n\}$. $\mathcal{S}$ wants to send a message m to a receiver in private and reliable

way. A computationally unbounded byzantine adversary $\mathcal{A}$ can control a subset of wires. The set of possible subsets of wires that the adversary can control forms the adversary structure Γ defined as

$$\Gamma = \{B \subset \mathcal{W} : \mathcal{A} \text{ can corrupt } B\}$$

We assume the adversary structure is monotone. This means that , if $C \in \Gamma$ and $C' \subset C \subset \mathcal{W}$, then $C' \in \Gamma$. A threshold adversary is a special case of an adversary structure where Γ includes all subsets of $\mathcal{W}$ with size at most t.

The receiver on the other hand can reconstruct the secret message using the values sent on specific subsets of wires, defined by the *access structure*, Σ defined as follows:

$$\Sigma = \{C \subset \mathcal{W} : \text{ values sent on wires in } C \text{ uniquely determine the message}\}$$

We assume Σ is monotone, that is, if $B \in \Sigma$ and $B \subset B' \subset \mathcal{W}$, then $B' \in \Sigma$. We assume any subset in $2^{\mathcal{W}}$ is either an adversary set or an access set. That is

$$\Gamma = 2^{\mathcal{W}} \setminus \Sigma$$

Definition 1. *An SMT protocol is called an $(0, \delta)$-Secure Message Transmission ($(0, \delta)$-SMT) protocol if the following conditions are satisfied:*

- ***Privacy:*** *$\mathcal{A}$ learns no information about the secret m. More precisely, if $\{w_{i_1}, ..., w_{i_k}\} \in \Gamma$ then for any $m \in \mathcal{M}$,*

$$\Pr(\mathcal{M} = m | X_{i_1} = x_{i_1}, ..., X_{i_k} = x_{i_k}) = \Pr(\mathcal{M} = m)$$

- ***General Reliability:*** *The receiver $\mathcal{R}$ always outputs $\hat{m} = m$ or* **NULL**. He never outputs an incorrect secret.
- ***Trivial Reliability:*** If the adversary blocks the transmissions on the wires he corrupts (*i.e.,* the received transmissions on those wires are all null strings) then the receiver $\mathcal{R}$ outputs $\hat{m} = m$.
- ***Failure:*** $\mathcal{R}$ receives the message m with probability $\geq 1 - \delta$. That is, if $\{w_{i_1}, ..., w_{i_k}\} \in \Sigma$ then

$$\Pr(\mathcal{R} \text{ outputs } \mathbf{NULL}) \leq \delta$$

When $\delta = 0$, we call it a **P**erfectly **S**ecure **M**essage **T**ransmission (**PSMT**, for short) or (0, 0)-SMT.

Comments: An alternative definition of general reliability used in the literature [12] allows the receiver to reconstruct a message $m' \neq m$. The trivial reliability requirement was first assumed in [14]. The requirement although natural, is not required in general.

Definition 2. *(Q^k condition [13]) An adversary structure Γ satisfies Q^k condition with respect to the wire set $\mathcal{W}$ if there are no k sets in Γ, which cover the full $\mathcal{W}$. Mathematically,*

$$Q^k \Leftrightarrow \forall B_1, ..., B_k \in \Gamma : B_1 \cup ... \cup B_k \neq \mathcal{W}$$

It has been shown that 1-round (0, 0)-SMT and 1-round (0, δ)-SMT is possible if and only if the adversary structure Γ satisfies the Q^3 [10] and Q^2 [18] conditions, respectively.

A 1-round (0, δ)-SMT tolerating a generalized adversary satisfying the Q^2 condition consists of a pair of algorithms (**Enc, Dec**) defined as follows.

- **Enc** is a probabilistic encoding algorithm which takes a secret $m \in \mathcal{M}$ as input and outputs an encoding $(x_1, ..., x_n)$, of the message. x_i is transmitted through the i^{th} wire. Enc is called by the sender $\mathcal{S}$.
- **Dec** is a deterministic decoding algorithm which takes $(x'_1, ..., x'_n)$, a corrupted version of the encoded message, and outputs $m' \in \mathcal{M}$ or **NULL** (denoting failure). **Dec** is called by the receiver $\mathcal{R}$.

$(x_1, ..., x_n)$ is corrupted to $(x'_1, ..., x'_n)$ by the adversary who controls wires corresponding to an adversary set. It is required that **Dec**(**Enc**(m))=m, for any $m \in \mathcal{M}$. Let X_i denote the random variable representing value x_i, and $\mathcal{X}_i$ denote the set of possible values $x_i, 1 \leq i \leq n$.
The number of ***rounds*** of a protocol is the number of interactions between $\mathcal{S}$ and $\mathcal{R}$. We consider synchronous network where time is divided into clock ticks and in each clock tick the sender or the receiver sends a message and the message is received by the other party before the next clock tick.
Communication complexity is the total number of bits transmitted (b) between $\mathcal{S}$ and $\mathcal{R}$ for communicating the message(s). Communication efficiency is often measured in terms of *transmission rate*, which is the ratio of the communication complexity to the length of the message $M_{\mathcal{S}}$. That is,

$$\textit{Transmission Rate} = \frac{\text{total number of field elements transmitted}(b)}{\text{size of the secrets}(|M_{\mathcal{S}}|)}$$

The message $M_{\mathcal{S}}$ is either one element or a sequence of elements from an alphabet.
Computation complexity is the amount of computation performed by $\mathcal{S}$ and $\mathcal{R}$ throughout the protocol. A protocol which needs *exponential* (in the size of the adversary structure) computation is called *inefficient*. Efficient protocols need *polynomial* (in the size of the adversary structure) computation.

Related Work. Chowdury, Kurosawa, and Patra gave an efficient 1-round (0, δ)-SMT protocol for against a generalized adversary satisfying the Q^2 condition [8].

2.2 Secret Sharing

A secret sharing scheme is a cryptographic primitive that distributes a secret s among n participants such that only qualified subsets of participants can reconstruct the secret, while non-qualified subsets get no information about the secret. In a secret sharing scheme, there are n participants $P = \{P_1, ..., P_n\}$ and a trusted dealer D. The set of participants who are qualified to reconstruct the secret is specified by an access structure $\Sigma \subseteq 2^P$. We consider monotone access structures in which any subset that contains a qualified subset is also a qualified

set. In this case the access structure Σ is uniquely determined by the family of minimal qualified subsets, Σ_0, known as the basis of Σ. The set of participants who are not allowed to learn any information about the secret is specified by the adversary structure Γ, which is defined as $\Gamma = 2^P \setminus \Sigma$. For monotone access structures, the adversary structure is also monotone. In a monotone adversary structure if $B \in \Gamma$ and $B' \subset B \subset P$, then $B' \in \Gamma$.

A secret sharing scheme with a monotone access structure Σ must satisfy the following conditions

- **SS1:** A set of participants $P_{i_1}, ..., P_{i_k}$ can reconstruct the secret if and only if $\{P_{i_1}, ..., P_{i_k}\} \in \Sigma$.
- **SS2:** Any set of participants $P \notin \Sigma$ must have no information about the secret.

A secret sharing model consists of two algorithms: **ShareGen** and **Reconst**. The share generation algorithm **ShareGen** takes a secret $s \in S$ as input and outputs a list $(v_1, v_2, ..., v_n)$. Each v_i is called a share and is given to participant P_i. The **ShareGen** algorithm is invoked by the dealer D. The secret reconstruction algorithm Reconst takes a list of shares and outputs a secret $s \in S$.

Let V_i denotes the random variable that represents the share values v_i and denote, $\mathcal{V}_i = \{v_i | \Pr[V_i = v_i] > 0\}$, the set of possible shares held by participant P_i.

The efficiency of a secret sharing scheme is measured by the *information rate* which is the ratio of the size of the secret to the size of the largest share given to any participant. The maximum possible rate is 1 and such scheme are called *ideal.*

2.3 Linear Secret Sharing Scheme

A secret sharing scheme for a monotone access structure Σ can be realized by a linear secret sharing scheme (LSSS). Let M be a $d \times e$ matrix over a finite field S and $\psi : \{1, ..., d\} \to \{1, ..., n\}$ be a labeling function, where $d \geq e$ and $d \geq n$. The share generation algorithm **ShareGen** is as follows.

ShareGen

1. To share a secret $s \in S$, the dealer D first chooses a random vector $\mathbf{r} \in S^{e-1}$ and computes a vector
$$\mathbf{v} = (v_1, ..., v_d)^T = M \times \begin{pmatrix} s \\ \mathbf{r} \end{pmatrix}$$
2. Let LSSS$(s, \mathbf{r})$= $(share_1, ..., share_n)$, here $share_i = \{v_j : \psi(j) = i\}$. P_i will receive $share_i$ as its share, $i = 1, ..., n$.

The reconstruction algorithm **Reconst** is as follows:

A set of participants $A \in \Sigma$ can reconstruct the secret s if and only if (1, 0, ..., 0) is in the linear span of
$$M_A = \{m_j : \psi(j) \in A\},$$
here m_j is the j^{th} row of M.

Definition 3. *(Monotone Span Program (MSP) [4]) The above* (M, ψ) *is a monotone span program which realizes* Σ*. The size of the MSP is the number of rows* d *in* M*.*

3 Secret Sharing Scheme with Cheaters

Tompa and Woll first considered the problem of secret sharing with cheaters [21] who submit wrong shares with the aim of learning the secret while preventing an honest member of a qualified set to do so. That is, given a qualified set $\{P_{i_1}, ..., P_{i_{k+1}}\} \in \Gamma$, the adversary can corrupt $P_{i_1}, ..., P_{i_k}$ (this is a non-qualified set) who will submit incorrect shares to cheat the $(k+1)^{st}$ participant. The cheaters (i.e., the adversary) succeed if the shares presented by $P_{i_1}, ..., P_{i_{k+1}}$ construct a secret $s' \neq s$. Secret sharing schemes that detect cheating has been studied by a number of authors using slightly different models. In [5], authors assumed that the cheaters know the secret when cheating, while some other [16] assumed the cheaters do not know the secret. Also in some cases it is assumed that the secret is uniformly distributed, while in other arbitrary distribution is assumed [15]. In nearly all cases the access structure is assumed threshold. Secret sharing schemes with cheating for a special type of access structure has been considered in [19, 17, 6]. To the best of our knowledge, only [15] considers non-threshold adversary for any monotone access structure.

In a secret sharing scheme with cheaters, the reconstrcution algorithm is modified to as follows. The secret reconstruction algorithm Reconst takes a list of shares corresponding to an access set and outputs either a secret $s \in S$, or a special symbol $\perp, \perp \notin S$. Here $\perp$ is a special symbol indicating the event that a cheating has been detected.

The success probability of the adversary in cheating is defined as $\Pr[s' \in S \wedge s' \neq s]$.

4 Relations between Secret Sharing with cheating and 1-round (0, δ)-SMT Against a Generalized Adversary

The relationship between 1-round (0, δ)-SMT with a threshold adversary structure, and threshold secret sharing scheme with cheating probability λ, has been considered in [14] where authors showed equivalence of the two under certain restrictions on the two. In this paper we revisit the same problem, assuming general adversary structure and derive restrictions under which the two are equivalent.

4.1 From Secret Sharing to Secure Message Transmission Tolerating a Generalized Adversary

Theorem 1. *If there exists a secure secret sharing scheme with cheating probability* $\leq \lambda$ *for the secret space* S *against a generalized adversary, then there exists a 1-round (0,* δ*)-SMT protocol against the same adversary satisfying the*

Q^2 condition for the message space $\mathcal{M} = S$ such that $\delta = \lambda(|\Sigma| - 1)$. Further it holds that $\mathcal{V}_i = \mathcal{X}_i$, for $1 \leq i \leq n$.

Proof:
We show a construction of SMT from secret sharing with cheating detection such that δ for the SMT is bounded by a function of λ in the secret sharing with cheating detection.

For a *maximal* adversary set $B = \{P_{i_1}, ..., P_{i_k}\} \in \Gamma$, the chance of cheating a target participant $P_{i_{k+1}}$ such that $C = \{P_{i_1}, ..., P_{i_k}, P_{i_{k+1}}\} \in \Sigma$ is at most λ, where cheating means that the reconstruction algorithm of secret sharing outputs a different secret s', than the original secret s.

Enc and **Dec** for SMT are generated from **ShareGen** and **Reconst** of secure secret sharing with cheating as follows. **Enc** is just the same as **ShareGen**. That is, on input of the secret message $m \in \mathcal{M}$, **Enc** runs **ShareGen** to generate $(x_1, ..., x_n) = (v_1, ..., v_n)$. Then $\mathcal{S}$ sends the shares over the wires, one share per wire.

Dec works as follows. Note that an adversary set can intersect with a number of access sets and so access sets will include corrupted members (cheaters). **Dec** invokes **Reconst** for every access set $a \in \Sigma$. The result will be a set $\{m, m_1, ..., m_i, \perp\}$ where $m_i \neq m$ corresponds to the output of **Reconst** when an access set includes cheaters and cheater detection algorithm has failed to detect the cheating. **Dec** algorithm outputs m if the output set is $\{m, \perp\}$, and **NULL** indicating failure, otherwise.

We have to show that all the conditions of a SMT are satisfied. The privacy of SMT follows from the corresponding property **SS2** of secret sharing with cheating. This is true as the participants corresponding to an adversary set in secret sharing with cheating learn no information about the secret.

Trivial reliability follows from the Q^2 property of the adversary structure. If the adversary blocks the transmission on the wires corresponding to an adversary set $\gamma \in \Gamma$, the remaining wires in $\sigma = \mathcal{W} \setminus \Gamma$, constitutes an access set with no cheating participant. The **Dec** algorithm uses **Reconst** on all the access sets. **Reconst** will output the correct m for σ. For all other access sets that are disjoint with γ, the same m will be output. For the access sets that have nonempty intersection with γ, there will not be sufficient shares and so **Reconst** will output $\perp$. So the set of messages that are output from **Reconst** will include only m and $\perp$ and so **Dec** will output m.

The message recovery algorithm **Dec** of SMT will output **NULL**, or the message m and will never output an incorrect message. This is because if the adversary corrupts an adversary set γ, its complement will be an access set and so **Reconst**, when applied to this access set, will result in m. According to the description of **Dec**, if the set of messages that are obtained from applying **Reconst** to all access sets include only this message (other than $\perp$) then this message will be output; otherwise **Dec** outputs **NULL**. Suppose the adversary corrupts an adversary set γ. All the access sets that have non-empty intersection with this set will include cheaters and so the **Reconst** algorithm applied to such sets can potentially output a message $m' \neq m$. The adversary will maximize

his success chance in causing **Dec** to fail, by choosing an adversary set that intersects with the most number of access sets. Note that since the complement of an adversary set is an access set, the maximum number of access sets that have nonempty intersection with γ is $|\Sigma| - 1$. For each such access sets the cheaters succeed in cheating with probability at most λ and so with probability at least $1-\lambda$ the cheating will be detected. **Dec** outputs the correct m if all the cheatings are detected. This will happen with probability at least $(1-\lambda)^{|\Sigma|-1} = 1-(|\Sigma|-1)\lambda$. That is $1-\delta \geq 1-(\Sigma|-1)\lambda$ or $\delta \leq -(|\Sigma|-1)\lambda$. □

4.2 From Secure Message Transmission to Secret Sharing Tolerating a Generalized Adversary

First consider the followings which hold if Q^2 holds.

1. The complement of an adversary set γ is an access set.
   This is true because a subset in 2^P is either an access set or an adversary set. Let $\bar{\gamma} = \mathcal{W} \setminus \gamma$. Then if $\bar{\gamma}$ is not an access set, it will be an adversary set which cannot be the case because of Q^2.
2. Complement of an access set σ is not necessarily an adversary set. It may be another access set.

In the following we assume that:

R1 The adversary structure satisfies Q^2.
R2 Complement of a minimal access set is a maximal adversary set.

Note that these requirements imply the following.
R3 The complement of a maximal adversary set is a minimal access set.

This is true because of the following. Suppose γ is a maximal adversary set. Then, because of Q^2 its complement is not an adversary set and so is an access set σ. Note that σ must be minimal. If not, then $\sigma' \subset \sigma$ is a minimal access set and by the above assumption (*R2*) $\gamma' = \mathcal{W} \setminus \sigma'$ is a maximal adversary set. We have $\gamma \subset \gamma'$ which contradicts with γ being maximal. With the above assumptions, we denote the access structures as $\hat{Q}^2$.

For a maximal adversary set γ, we use σ to represent the set $\bar{\gamma} = \mathcal{W} \setminus \gamma$ as defined above, i.e., $\bar{\gamma} = \sigma \in \Sigma$. Note that σ is a minimal access set. For $\sigma = \{i_1, \cdots i_k\}$, define the function $F_\sigma(x'_{i_1}, ..., x_{i_k})$ as follows:

$$F_\sigma(x'_{i_1}, ..., x'_{i_k}) = m_\sigma \text{ or } \perp, \quad (w_{i_1}, ..., w_{i_k}) \in \Sigma$$

were, m_σ is defined as follows:

$m_\sigma =$ the unique message that because of trivial reliability, is the output of the decoding algorithm when wires in γ are blocked,
$=$ for other (not blocking) corruptions, either (i) a message m' if this is the unique message that has $x'_{i_1}, ..., x'_{i_k}$ on these wires, or
(ii) $\perp$ if more than one message has $x'_{i_1}, ..., x'_{i_k}$ on these wires.

Note that the above function*is defined for all minimal access sets* because from the assumption on the adversary set, any minimal access set is the complement of a maximal adversary set.

We say that a 1-round (0, δ)-SMT protocol is *canonical* if, for all corrupted transcripts $(x'_1, x'_2, \cdots x'_n)$ of a message m (from the original transcript $(x_1, x_2, \cdots, x_n)$ corrupted by an adversary set) we have:

$$\begin{aligned}\mathbf{Dec}(x'_1, ..., x'_n) = m, \text{ if } F_\sigma(x'_{i_1}, ..., x'_{i_k}) = m, \text{ or } \perp \text{ for all minimal access sets } \sigma \in \Sigma \\ = \mathbf{NULL}, otherwise \end{aligned} \tag{1}$$

where $m \in \mathcal{M}$.

That is decoding function of a canonical SMT can be written in terms of $F_\sigma()$ for all minimal access sets $\sigma \in \Sigma$.

Theorem 2. *If there exists a canonical 1-round (0, δ)-SMT protocol for the message space $\mathcal{M}$, then there exits a secure secret sharing scheme with cheating probability $\leq \lambda$ where $\lambda \leq \delta$, for the secret space $S = \mathcal{M}$. Further it holds that $\mathcal{X}_i = \mathcal{V}_i$, for $1 \leq i \leq n$.*

Proof:
The share generation algorithm **ShareGen** works by invoking **Enc** of SMT as follows. On input a secret $s \in S$, **ShareGen** calls **Enc**(s) and generates $(v_1, ..., v_n) = (x_1, ..., x_n)$. The Dealer D gives the share x_i to the participant $P_i, 1 \leq i \leq n$.

We have to show that the conditions **SS1** and **SS2** of secret sharing with cheating are satisfied. **SS1** requires that for any access set there is a unique message that can be constructed from the set of participants' shares. An access set contains a minimal access set which because of trivial reliability, uniquely determines the secret. **SS2** follows from the privacy condition of SMT. This is true because in SMT the shares corresponding to an adversary set, reveal no information about the secret.

To show that success probability of cheaters in the secret sharing scheme is less than δ, we assume this is not true. That is there is a *maximal* adversary set $\gamma = \{w_{i_1}, \cdots, w_{i_k}\}$, such that cheaters using the shares on $\{w_{i_1}, \cdots, w_{i_k}\}$ can cheat $P_{i_{k+1}}$ with probability higher than δ.

Now suppose in the SMT the adversary corrupts wires $\{w_{i_1}, \cdots, w_{i_k}\}$. The SMT decoding algorithm is canonical and so uses $F_\sigma()$ for all minimal access sets as defined in decoding function definition in (1). Note that the set $\{w_{i_1}, \cdots, w_{i_k}, w_{i_{k+1}}\}$ is a minimal access set (this is true because $\{w_{i_1}, \cdots, w_{i_k}$ is a maximal adversary set and so adding $w_{i_{k+1}}$ will make it an access set. This access set is minimal also- because if it is not, then a subset of it is a minimal access set which implies that γ is not maximal.) and $F_{\sigma_1}()$ will output $m' \neq m$ with probability higher than δ. On the other hand secret reconstruction algorithm applied to the complement of γ which is a minimal access set results in a message m and so the $Dec(x'_1, \cdots, x'_n)$ will output $\perp$ with probability higher than δ. □

5 Lower Bound on Communication Complexity for 1-round (0, 0)-SMT and (0, δ)-SMT Against a Generalized Adversary

Using the equivalence between canonical (0, δ)-SMT and secret sharing with cheating against a generalized adversary, and the known bounds on share size of the latter, we can derive lower bound on the communication complexity of canonical (0, δ)-SMT against a generalized adversary (using our definition of reliability).

Proposition 1. *The lower bound on the communication complexity of a canonical 1-round (0, δ)-SMT tolerating a generalized adversary with adversary structure Γ and satisfying the condition Q^2 is $n \log(\frac{|\mathcal{M}|-1}{\delta}+1)$, where δ is the cheating probability and $\mathcal{M}$ is the message space.*

Proof: A lower bound on the share size of secret sharing scheme with cheating probability δ is $\log(\frac{|\mathcal{M}|-1}{\delta}+1)$, derived in [16]. The result follows by noting that the secret sharing scheme can be used to construct a canonical (0, δ)-SMT (See proof of Theorem 1).

In the following we will derive a lower bound on the communication complexity of 1-round (0, 0)-SMT tolerating a generalized adversary,

We will first state the following lemma.

Lemma 1. *In a 1-round (0, δ)-SMT tolerating a generalized adversary with $\delta < 1/2$, for any pair of adversary sets B_i and B_j, the information transmitted on the wires in the set $C_{ij} = [n] - (B_i \cup B_j)$ will uniquely determine the message. Here C_{ij} is the set of all wires minus the wires in B_i and B_j.*

This is true because otherwise, we have two different messages m and m' such that:
–XYZ is the transmission over the wires in C, B_i, B_j, respectively when message m is sent;
–$XY'Z'$ is the transmission over the wires in C, B_i, B_j, respectively when message m' is sent.

Now the transcript $XY'Z$ corresponds to message m with adversary set B_i and m' with adversary set B_j, respectively. So the success chance of the receiver in correctly outputting the message is 1/2.

Theorem 3. *A 1-round PSMT protocol tolerating a generalized adversary with the adversary structure Γ satisfying Q^3, has communication complexity lower bounded by $\geq \frac{\gamma}{\gamma-3}\ell$, where $\ell = \log|\mathcal{M}|$ is the size of the message in bit and $\gamma = |\Gamma|$.*

Proof. The proof of the theorem is inspired by the proof in [11] for threshold adversary.

Let T_i^m denote the set of all possible transmissions that can occur on wire $w_i \in \mathcal{W}$ when the sender $\mathcal{S}$ transmits message m. Suppose for $j \geq i$, $M_C^m \subseteq$

$\times_{w_i \in C} T_i^m$ denote the set of all possible transmissions (as vectors) that can occur on the set of wires in the set C when $\mathcal{S}$ transmits message m. Let the set of all transmission (for all messages) on wire w_i given by, $T_i = \bigcup_{m \in M} T_i^m$ be the capacity of the wire w_i. The capacity of the set C of wires is $M_C = \bigcup_{m \in M} M_C^m$.

Perfect reliability of a SMT protocol implies that the (uncorrupted) transmissions on the wires in any set C_{ij}, uniquely determines the secret (Lemma 1) and so:

$$M_{C_{ij}}^{m_1} \cap M_{C_{ij}}^{m_2} = \emptyset \tag{2}$$

On the other hand, perfect privacy means that if the adversary $\mathcal{A}$ corrupts any set in Γ, they should get not have any information about the message m. This implies the transmission vector corresponding to the adversary set corrupted by the adversary, should reveal no information about the message and so is a possible transmission for any message. Thus for any two messages $m_1, m_2 \in M$ and any adversary set $B \in \Gamma$, it must be true that

$$M_B^{m_1} = M_B^{m_2} \tag{3}$$

We know that for a 1-round $(0,0)$-SMT, Γ satisfies Q^3.

Let $|\Gamma| = \gamma$ and define $C_{ijk} = [n] - (B_i \cup B_j \cup B_k)$. We know that the wires in C_{ij} uniquely determine the secret. Also that for any $B_i \cup B_j \cup B_k$, transmission on $B_i \cup B_j \cup B_k \setminus B_i \cup B_j$ will be common to all messages (it is a subset of transmission in M_{B_k}), and so transmission on wires in C_{ijk} will satisfy, $|C_{ijk}| \geq |M|$.

And so,

$$\prod_{a \in C_{ijk}} |T_a| > |M_{C_{ijk}}| > |M| \tag{4}$$

Now consider the following product:

$$\prod_{B_i,B_j,B_k \in \Gamma} \prod_{a \in C_{ijk}} |T_a| > \prod_{B_i,B_j,B_k \in \Gamma} M_{C_{ijk}} > |M|^{\frac{\gamma(\gamma-1)(\gamma-2)}{6}} \tag{5}$$

where $\binom{\gamma}{3} = \frac{\gamma(\gamma-1)(\gamma-2)}{6}$ is the number of C_{ijk}.

We would like to find the maximum number of times that a wire can appear in $\prod_{B_i,B_j,B_k \in \Gamma} \prod_{a \in C_{ijk}} |T_a|$.

Note that a wire x must appear in *at least* one adversary set (otherwise that wire is secure and can be used for secure transmission). This means that at most $\binom{\gamma-1}{3} = \frac{(\gamma-1)(\gamma-2)(\gamma-3)}{6}$ sets of the form C_{ijk} will include the wire x.

This means that we can always (for any Γ) write,

$$(\prod_{a=1}^{n} |T_a|)^{\frac{(\gamma-1)(\gamma-2)(\gamma-3)}{6}} \geq \prod_{B_i,B_j,B_k \in \Gamma} \prod_{a \in C_{ijk}} |T_a| > \prod_{B_i,B_j,B_k \in \Gamma} M_{C_{ijk}} > |M|^{\frac{\gamma(\gamma-1)(\gamma-2)}{6}} \tag{6}$$

That is,

$$(\gamma - 3)\log \Sigma_{a=1}^{n}|T_a|) \geq \gamma \log |M|$$

and,

$$\frac{\Sigma_{a=1}^{n}\log |T_a|)}{\log |M|} \geq \frac{\gamma}{\gamma - 3}$$

Therefore, the lower bound on the transmission rate of a 1-round (0, 0)-SMT against a generalized adversary is $\frac{\gamma}{\gamma-3}$.

Thus the lower bound on the communication complexity is $\geq \frac{\ell\gamma}{\gamma-3}$. □

6 An Efficient 1-round (0, δ)-SMT Protocol Against a Generalized Adversary

In this section we construct an efficient 1-round (0, δ)-SMT protocol against a generalized adversary inspired by a construction of cheater detecting ε-secure secret sharing schemes [15]. The construction uses linear secret sharing scheme (LSSS) and a special class of hash functions.

The basic idea of the protocol is to generate a key-dependent hash value for the message, and then send the shares of the message and the key of the hash function, generated using two LSSSs, over the wires. The hash value of the message is broadcasted to the receiver. The receiver will be able to recover the message with small failure probability, from the received shares and hash value.

The hash functions required for this construction are called, *strongly key-differential universal hash function*, introduced in [15].

Definition 4. *A family of hash function $H : \mathcal{A} \to \mathcal{B}$ is called a strongly key-differential unviersal ε-SKDU$_2$ if there exists $\hat{b} \in \mathcal{B}$ such that for two distinct $a, a' \in \mathcal{A}$ and for any $c \in \mathcal{E}$,*

$$\frac{|\{h_e | e \in \mathcal{E}, h_e(a) = \hat{b}, h_{e+c}(a') = \hat{b}\}|}{|\{h_e | e \in \mathcal{E}, h_e(a) = \hat{b}\}|} \leq \varepsilon$$

such that for any $a \in \mathcal{A}$ and $b \in \mathcal{B}$, $|\{h_e \in H | h_e(a) = b\}| = \frac{|H|}{|\mathcal{B}|}$

Construction Let H be a family of ε-SKDU$_2$ that maps the set $A = \mathcal{M}$ to B.

Let $\mathtt{SS}_1 = (ShareGen_1, Reconst_1)$, and $\mathtt{SS}_2 = (ShareGen_2, Reconst_2)$, be two linear secret sharing scheme for adversary set Γ, for the message set $\mathcal{M}$ and key set $\mathcal{E}$, respectively. and assume Γ satisfies Q^2. Let $\Sigma = 2^P \setminus \Gamma$.

The SMT protocol Π is described below.

– **Message Transmission:**
Suppose the sender $\mathcal{S}$ wants to send a secret $m \in \mathcal{M}$ to the receiver $\mathcal{R}$. $\mathcal{S}$ does the following.

Step 1 Randomly selects $e \in \mathcal{E}$ and find $h_e(m) = \hat{b}$.

Step 2 Calls **ShareGen**$_i$ of $\mathtt{SS}_i, i = 1, 2$, to generate two sets of shares for m and e respectively: $(s_1^m, ..., s_n^m) \leftarrow \mathtt{SS}_1(m)$, and $(s_1^e, ..., s_n^e) \leftarrow \mathtt{SS}_2(e)$

Step 3 sends $s_i = (s_i^m, s_i^e), \hat{b}$ through wire $i, 1 \leq i \leq n$.

– **Message Recovery:**
The receiver $\mathcal{R}$ receives the shares $s'_i = (s_i'^m, s_i'^e)$ through wire $i, 1 \leq i \leq n$. $\mathcal{R}$ and does the following.

Step 1 Set $\mathcal{L} = \emptyset$. For each minimal access set $a \in \Sigma$ do:
1. Call the reconstruction algorithms **Reconst**$_i, i = 1, 2$, on the shares $\{s_i'^m | i \in a\}$ and recover $\hat{m}$ and $\hat{e}$.
2. If $h_{\hat{e}}(\hat{m}) \neq \hat{b}$, output **NULL**. Otherwise add $\hat{m}$ to $\mathcal{L}$.

Step 2 If $\mathcal{L}$ contains more than one distinct value, output **NULL**. Otherwise output the unique element in $\mathcal{L}$, as the protocol output.

Let k denote the number of minimal access sets in Σ.

Theorem 4. *Π is a 1-round (0, δ)-SMT protocol, $\delta = (k-1)\epsilon$ tolerating the generalized adversary with the adversary structure Γ satisfying Q^2. The protocol sends a message that is one field element, by transmitting $O(n)$ field elements.*

Proof (sketch):
Perfect privacy:
Suppose the adversary corrupts a maximal adversary set B and accesses the values sent over the associated wires. Let $B = \{w_{i_1} \cdots, w_{i_{t-1}}\}$ with message and key shares, $s_{i_j} = (s_{i_j}^m, s_{i_j}^e), j = 1, \cdots, t-1$. We need to show that $\Pr(m | s_{i_j}, j = 1, \cdots, t-1, h_e(m) = \hat{b}) = \Pr(m)$. Note that because of perfect secrecy of $\mathtt{SS}_i, i = 1, 2$, and independent choice of m and e, we have $\Pr(m | s_{i_j}, j = 1, \cdots, t-1) = \Pr(m)$ (also $\Pr(e | s_{i_j}, j = 1, \cdots, t-1) = \Pr(e)$). For any $\hat{b}$, we have $\Pr(m, e | h_e(m) = \hat{b})$ is the number of pairs e, m where $h_e(m) = \hat{b}$ divided by $|\mathcal{M}| \times |\mathcal{E}|$ which because of the property of hash function, is constant and so $\hat{b}$ does not leak any information about the pair.
δ-reliability:
Receiver attempts to recover the message for all minimal access sets. Note that because of the Q^2 property, for any maximal adversary set B that is corrupted by the adversary, the set $\mathcal{W} \setminus B$ is an access set (otherwise it is an adversary set which contradicts Q^2) and contains a minimal access set. So the correct message will be reconstructed for this access set and so $\mathcal{L}$, will always contain the correct message.

Note that the adversary may succeed in constructing a pair of message and key, m', e', such that $h_{e'}(m') = \hat{b}$. The following argument shows that this probability is at most ϵ.

This is true because of the following. Let $B = \{w_{i_1}, \cdots, w_{i_{t-1}}\}$ with message and key shares, $s_{i_j} = (s_{i_j}^m, s_{i_j}^e), j = 1, \cdots, t-1$. Note that due to linear property of secret sharing schemes that are recombination constants $c_{A,j}^M, j = 1, \cdots, t-$

$1, x$, and $c^E_{A,j}, j = 1, \cdots, t_1, x,$, that depend on the access set $A = B \cup \{x\}$ such that

$$m = c^M_{A,x} s^m_x + \sum_{i=1}^{t-1} c^M_{A,j} s^m_{i_j}$$
$$e = c^E_{A,x} s^e_x + \sum_{i=1}^{t-1} c^E_{A,j} s^e_{i_j}$$

Now e' and m' are constructed using the same constants, but forged values of shares for B and so,

$$m' = c^M_{A,x} s^m_x + \sum_{i=1}^{t-1} c^M_{A,j} s'^m_{i_j}$$
$$e' = c^E_{A,x} s^e_x + \sum_{i=1}^{t-1} c^E_{A,j} s'^e_{i_j} \Rightarrow e' = e + \sum_{i=1}^{t-1} (c^E_{A,j} s'^e_{i_j} - c^E_{A,j} s^e_{i_j}) = e + C$$

The success probability of the adversary is, $\Pr(m' \in \mathcal{M}, m' \neq m, h_{e'}(m') = \hat{b})$ which is bounded as,

$$\Pr(m' \in \mathcal{M}, m' \neq m, h_{e'}(m') = \hat{b} | h_e(m) = \hat{b})$$
$$\frac{\Pr(h_e(m) = \hat{b}, h_{e+C}(m') = \hat{b})}{\Pr(h_e(m) = \hat{b})} = \frac{|\{h_e : h_e(m) = \hat{b}, h_{e+C}(m') = \hat{b}\}|}{|\{h_e : h_e(m) = \hat{b}\}|} \leq \epsilon$$

Note that according to the SMT reconstruction algorithm, adversary's success in this case result in the SMT protocol to output **NULL**. Then, at most $k-1$ can be influenced by the adversary. Probability of protocol not outputting **NULL** is at least $(1-\epsilon)^{k-1} \approx 1-(k-1)\epsilon$. That is $\delta = (k-1)\epsilon$.

6.1 Comparison with the protocol in [8]

Chowdhury, Kurosawa, and Patra designed an efficient 1-round $(0, \delta)$-SMT protocol tolerating a generalized adversary [8]. But their protocol needs to send $\ell = n$ messages. On the other hand, our protocol can work for a single message. In many scenarios we may need to send just one message, for example a key in sensor network. In those scenarios our protocol is better than their protocol. It is to be noted here that both the protocols needs computation which is polynomial in the size of the adversary structure Γ and the underlying LSSS.

References

1. T. Araki. Almost Secure 1-Round Message Transmission Scheme with Polynomial-time Message Decryption. In Proc. of ICITS 2008.

2. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness Theorems for Non-cryptographic Fault-tolerant Distributed Computation (extended abstract). In Proc. of STOC, pp. 1–10, 1988.
3. D. Chaum, C. Crepeau, and I. Damgard. Multiparty Unconditionally Secure Protocols (Extended Abstract). In Proc. of FOCS, pp. 11–19, 1988.
4. R. Cramer, I. Damgrd, U. Maurer. General Secure Multi-party Computation from any Linear Secret-Sharing Scheme. In Proc. of EUROCRYPT: 316-334 (2000)
5. M. Carpentieri, A. De Santis, and U. Vaccaro. Size of Shares and Probability of Cheating in Threshold Schemes. In Proc. of EUROCRYPT: 118-125 (1993)
6. S. Cabello, C. Padr, G. Sez. Secret Sharing Schemes with Detection of Cheaters for a General Access Structure. In Des. Codes Cryptography 25(2): 175-188 (2002)
7. C. Padr, G. Sez, and J. Villar. Detection of Cheaters in Vector Space Secret Sharing Schemes. In Des. Codes Cryptography 16(1): 75-85 (1999)
8. A. Choudhary, A. Patra, and K. Kurosawa. Simple and Efficient Single Round Almost Perfectly Secure Message Transmission Tolerating Generalized Adversary In Proc. of ACNS (2011)
9. D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly Secure Message Transmission. In Journal of the ACM, 40(1):17–47, 1993.
10. Y. Desmedt, Y. Wang, and M. Burmester. A Complete Characterization of Tolerable Adversary Structures for Secure Point-to-Point Transmissions Without Feedback. In Proc. of ISAAC: 277-287 (2005).
11. M. Fitzi, M. Franklin, J. Garay, H. Vardhan. Towards Optimal and Efficient Perfectly Secure Message Transmission. In Proc. of TCC: 311-322 (2007)
12. M. K. Franklin and R. N. Wright. Secure Communication in Minimal Connectivity Models. In Journal of Cryptology, 13(1):9–30, 2000.
13. M. Hirt and U. Maurer. Player Simulation and General Adversary Structures in Perfect Multiparty Computation. J. Cryptology 13(1): 31-60 (2000).
14. K. Kurosawa and K. Suzuki. Almost Secure (1-round, n-channel) Message Transmission Scheme. In Proc. of ICITS, volume 4883 of LNCS, pages 99–112, 2009.
15. S. Obana and T. Araki. Almost Optimum Secret Sharing Schemes Secure Against Cheating for Arbitrary Secret Distribution. In Proc. of ASIACRYPT: 364-379 (2006)
16. W. Ogata, K. Kurosawa, and D. R. Stinson. Optimum Secret Sharing Scheme Secure against Cheating. In SIAM J. Discrete Math. 20(1): 79-95 (2006)
17. C. Padro. Robust Vector Space Secret Sharing Schemes. In Inf. Process. Lett. 68(3): 107-111 (1998)
18. A. Patra, A. Choudhary, K. Srinathan, and C. Rangan. Unconditionally Reliable and Secure Message Transmission in Undirected Synchronous Networks: Possibility, Feasibility and Optimality. In IJACT 2(2): pages 159–197, 2010.
19. C. Padr, G. Sez, and J. Villar. Detection of Cheaters in Vector Space Secret Sharing Schemes. In Des. Codes Cryptography 16(1): 75-85 (1999)
20. T. Rabin and M. Ben-Or. Verifiable Secret Sharing and Multiparty Protocols with Honest Majority (Extended Abstract). In Proc. of STOC, pp. 73–85, 1989.
21. M. Tompa and H. Woll. How to Share a Secret with Cheaters. In J. of Cryptology 1(2): 133-138 (1988)
22. Y. Wang Robust Key Establishment in Sensor Networks. In SIGMOD Record 33(1): 14–19, 2004.
23. J. Wu and Douglas R. Stinson Three Improved Algorithms for Multi-path Key Establishment in Sensor Networks Using Protocols for Secure Message Transmission. Available at http://eprint.iacr.org/2009/413.pdf.