# Simple and Efficient Single Round Almost Perfectly Secure Message Transmission Tolerating Generalized Adversary

Ashish Choudhury[1] ⋆, Kaoru Kurosawa[2], and Arpita Patra[3]

[1] Applied Statistics Unit
Indian Statistical Institute Kolkata India
partho_31@yahoo.co.in, partho31@gmail.com
[2] Department of Computer and Information Sciences
Ibaraki University, Hitachi, Ibaraki Japan
kurosawa@mx.ibaraki.ac.jp
[3] Department of Computer Science
Aarhus University, Denmark
arpitapatra10@gmail.com, arpita@cs.au.dk

**Abstract.** Patra et al. [25] gave a necessary and sufficient condition for the possibility of *almost perfectly secure message transmission* protocols[4] tolerating *general, non-threshold* $\mathcal{Q}^2$ adversary structure. However, their protocol requires *at least three* rounds and performs *exponential* (exponential in the size of the adversary structure) computation and communication. Moreover, they have left it as an open problem to design efficient protocol for almost perfectly secure message transmission, tolerating $\mathcal{Q}^2$ adversary structure.

In this paper, we show the first *single* round almost perfectly secure message transmission protocol tolerating $\mathcal{Q}^2$ adversary structure. The computation and communication complexities of the protocol are both *polynomial* in the size of underlying *linear secret sharing scheme* (LSSS) and adversary structure. This solves the open problem raised in [25].

When we restrict our general protocol to *threshold adversary* with $n = 2t + 1$, we obtain a single round, *communication optimal* almost secure message transmission protocol tolerating threshold adversary, which is much more *computationally efficient* and *relatively simpler* than the previous communication optimal protocol of [35].

**Keywords**: Information theoretic security, non-threshold adversary, Byzantine corruption, efficiency.

## 1 Introduction

Consider the following problem: there exists a sender **S** and a receiver **R**, who are part of a large distributed network and connected by $n$ disjoint channels.

⋆ Financial Support from Department of Information Technology, India Acknowledged

[4] The authors in [25] called this problem as *unconditionally secure message transmission* (USMT).

There exists a *computationally unbounded adversary*, who can listen and forge communication over some of these channels in any arbitrary manner. However, neither **S**, not **R** knows which channels are under the control of the adversary. **S** has a message $m^{\mathbf{S}}$, which is a sequence of $\ell$ elements from a finite field $\mathbb{F}$, where $\ell \geq 1$. The challenge is to design a protocol, such that after interacting with **S** as per the protocol, the following should hold at **R**'s end:

1. **Perfect Reliability**: **R** outputs $m^{\mathbf{R}} = m^{\mathbf{S}}$.
2. **Perfect Secrecy**: Adversary should not get any *extra* information about $m^{\mathbf{S}}$. In other words, $m^{\mathbf{S}}$ should be *information theoretically secure*.

Moreover, we require that the above conditions should hold, irrespective of the behavior of the adversary. This problem is called as *perfectly secure message transmission* (PSMT) [12].

### 1.1 Motivation and Different Models for Studying PSMT

PSMT is a well known and fundamental problem in secure distributed computing. If **S** and **R** are directly connected by a secure channel, as assumed in generic *multiparty computation* (MPC) protocols [3, 4, 29] , then PSMT is trivial. However, if **S** and **R** are not directly connected by a secure channel, then PSMT protocols help to simulate a *virtual secure channel* between **S** and **R**. The second motivation for PSMT is to achieve information theoretic security. The security of all existing public key cryptosystems is based on hardness assumptions of certain number theoretic problems and security of these schemes hold only against a computationally bounded adversary. However, with the advent of new computing paradigms like Quantum computing [33] and with increase in computing speed, these assumptions may tend to be useless. But all these factors have no affect on PSMT protocols, as security of these protocols holds good against a computationally unbounded adversary.

Over the past two decade, PSMT problem has been studied by several researchers in different settings. Specifically, we can consider the following settings:

1. **Type of Channels**: The channels between **S** and **R** can be *bi-directional*. This setting has been considered in [12, 31, 16, 36, 1, 13, 19, 26, 24]. On the other hand, channels may be *uni-directional*, having direction associated with them [10, 23, 21, 40, 6].
2. **Adversary Capacity**: The adversary may be characterized by a *threshold*, say $t$, such that adversary can control *any* $t$ out of the $n$ channels [12, 31, 36, 19] or the adversary may be characterized as a more general *non-threshold* adversary, specified by an adversary structure [16, 28, 40, 41, 17, 18].
3. **Adversary Behavior**: The adversary may be *static* who corrupts the same channels throughout the protocol [12, 31, 36, 19] or the adversary may be *mobile*, who corrupts different set of channels, during different stages of the protocol [39, 26, 5, 27].
4. **Type of Underlying Network**: The underlying network may be *synchronous*, where there is a global clock in the system and the delay in the

transmission over any channel is bounded by a constant [12, 31, 36, 38, 19] or the network may be *asynchronous*, having no such global clock [30, 6]

Any PSMT protocol is analyzed by the following parameters:

1. **Round Complexity**: It is the number of communication rounds taken by the protocol, where a round is a communication from **S** to **R** or vice-versa.
2. **Communication Complexity**: It is the total number of field elements sent by **S** and **R** in the protocol.
3. **Computational Complexity**: It is amount of computation which is done by **S** and **R** in the protocol.

A PSMT protocol is said to be *efficient*, if the round complexity, communication complexity and computational complexity of the protocol is *polynomial* in $n$ and size of the *adversary structure* (see Sec. 1.2 for details about adversary structure). Irrespective of the settings in which PSMT problem is studied, the following questions are fundamental:

– **Possibility**: What are the necessary and sufficient conditions for the existence of any PSMT protocol, tolerating a given type of adversary?
– **Feasibility**: Once the possibility of a protocol is ascertained, the next obvious question is whether there exists an *efficient* protocol or not?
– **Optimality**: Given a message of some specific length, what is the lower bound on the round complexity and communication complexity of any PSMT protocol to send the message? Moreover, do we have a protocol, whose total round complexity and communication complexity matches these bounds?

Different techniques are used to answer the above questions in different settings. For details, see [7]. The issue of **Possibility, Feasibility** and **Optimality** of PSMT has been completely resolved tolerating *threshold adversary*. However, not too much is known regarding the **Feasibility** and **Optimality** of protocols against non-threshold adversary (see [7] for complete details) .

### 1.2 Non-Threshold Adversary

Let the set of $n$ channels be denoted by $\mathcal{W} = \{w_1, \ldots, w_n\}$. Then a *threshold adversary* is characterized by a threshold $t$, such that adversary can control any $t$ channels out of the $n$ channels for corruption. We denote such an adversary by $\mathcal{A}_t$. On the other hand, a *non-threshold general adversary* $\mathcal{A}$ is characterized by an *adversary structure* $\Gamma$, which is a collection of subsets of channels that the adversary $\mathcal{A}$ can *potentially* corrupt. That is,

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

Moreover, we assume that if $B \in \Gamma$ and if $B' \subset B$, then $B' \in \Gamma$. It is easy to see that a threshold adversary is a special case of non-threshold adversary, such that $|B| \leq t$, for each $B \in \Gamma$.

**Definition 1 ($\mathcal{Q}^k$ Condition [15]).** *We say that $\mathcal{A}$ satisfies $\mathcal{Q}^k$ condition with respect to $\mathcal{W}$, if there exists no $k$ sets in $\Gamma$, which adds upto the whole set $\mathcal{W}$. That is:*

$$\forall B_1, \ldots, B_k \in \Gamma : B_1 \cup \ldots \cup B_k \neq \mathcal{W}.$$

### 1.3 PSMT Tolerating Non-Threshold Adversary

Modeling the adversary by a threshold helps in easy characterization of protocols and it also helps in analyzing protocols. However, as mentioned in [15], modeling the (dis)trust in the network as a threshold adversary is not always appropriate because threshold protocol requires more *stringent* requirements than the reality [16]. Motivated by this, Kumar et al. [16] studied PSMT tolerating non-threshold adversary, where they resolved the issue of **Possibility** of PSMT protocols, when the channels are *bi-directional*. Specifically, they showed that *two or more* round PSMT is possible iff $\mathcal{A}$ satisfies $\mathcal{Q}^2$ condition with respect to $\mathcal{W}$. Recently, [41] resolved the issue of **Feasibility** of *multi-round* PSMT by designing two round efficient PSMT protocol tolerating non-threshold adversary if $\mathcal{A}$ satisfies $\mathcal{Q}^2$ condition.

On the other hand, Desmedt et al. [11] have shown that *one round* PSMT is possible iff $\mathcal{A}$ satisfies $\mathcal{Q}^3$ condition with respect to $\mathcal{W}$. Moreover, they also presented a one round PSMT protocol tolerating non-threshold adversary. However, their protocol is not efficient in general. Recently, Kurosawa [17] resolved the issue of **Feasibility** of *one round* PSMT by designing efficient one round PSMT protocol tolerating non-threshold adversary if $\mathcal{A}$ satisfies $\mathcal{Q}^3$ condition.

The issue of **Possibility** and **Feasibility** of PSMT tolerating non-threshold adversary for the case when the channels are *uni-directional* is resolved in [28, 40, 41]. So in short, there exists efficient PSMT protocols tolerating non-threshold adversary for bi-directional channels [41, 17] as well as for uni-directional channels [41]. However, there exists another variant of PSMT, known as *almost perfectly secure message transmission* (almost-PSMT), which got relatively less attention in the context of non-threshold adversary.

### 1.4 Almost Perfectly Secure Message Transmission: almost-PSMT

In PSMT, it is required that **R** should output $m^{\mathbf{R}} = m^{\mathbf{S}}$ without any error. In [14], the authors considered a variant of PSMT called almost-PSMT, where they relaxed this requirement. Specifically, a protocol is called almost-PSMT, if it satisfies the following requirements:

1. **Perfect Secrecy**: Same as in the case of PSMT.
2. **Almost Perfect Reliability**: **R** outputs $m^{\mathbf{R}} = m^{\mathbf{S}}$ with probability at least $1 - 2^{-\Omega(\kappa)}$, where $\kappa$ is the error parameter and $\kappa > 0$.

In [14], the authors studied almost-PSMT tolerating threshold adversary and showed that almost-PSMT protocols requires less number of channels than PSMT protocols for tolerating a threshold adversary with the same threshold. *That is, allowing a negligible error probability in protocol outcome reduces the connectivity requirement.* The work of [14] is followed by [10, 37, 20, 35, 2, 9, 22] where almost-PSMT tolerating threshold adversary is studied rigorously and the issues related to the **Possibility, Feasibility** and **Optimality** of almost-PSMT tolerating threshold adversary has been completely resolved. In summary, all these works show that *allowing a negligible error probability in the protocol output (without compromising the secrecy) results in significant reduction in the round complexity, communication complexity and also connectivity requirement (number of channels) of PSMT protocols.*

*Remark 1.* (**On the Term almost-PSMT**): In the literature, almost-PSMT protocols are also known by various other names. In [34, 37], the authors called these protocols as *probabilistic PSMT* (PPSMT). On the other hand, [25, 35] called these protocols as *unconditionally secure message transmission* (USMT) protocols. Finally, [7] called these protocols as *statistically secure message transmission* (SSMT) protocols. However, all the above terms stand for almost-PSMT. In this article, we prefer to use the original name, namely almost-PSMT.

### 1.5 Almost-PSMT Tolerating Non-Threshold Adversary: Motivation of Our Work

Unlike almost-PSMT tolerating threshold adversary, almost-PSMT against non-threshold adversary has got very less attention. In [25], Patra et al. have studied almost-PSMT tolerating non-threshold adversary. They showed that *single round as well as multi-round almost-PSMT* is possible iff $\mathcal{A}$ satisfies $\mathcal{Q}^2$ condition. This is to be compared with the results of [11] and [16], according to which *single round* and *multi-round* PSMT is possible iff $\mathcal{A}$ satisfies $\mathcal{Q}^3$ and $\mathcal{Q}^2$ condition respectively. Unfortunately, the almost-PSMT protocol tolerating non-threshold adversary presented in [25] is very inefficient and requires computation and communication complexity, which is exponential in the size of adversary structure. Moreover, it requires at least three rounds. In [25], the authors have left it as an open problem to design efficient almost-PSMT protocol tolerating non-threshold adversary, satisfying $\mathcal{Q}^2$ condition. In this paper, we solve this open problem.

### 1.6 Our Results and Comparison with the Existing Results

In this paper, we present the first single round almost-PSMT protocol tolerating non-threshold adversary $\mathcal{A}$, specified by an adversary structure, satisfying $\mathcal{Q}^2$ condition. Our protocol is *round optimal*, requiring minimum number of rounds. Moreover, our protocol is very simple and efficient and thus significantly outperforms the almost-PSMT protocol of [25].

As a special case of our single round protocol, when we restrict it to *threshold adversary*, we get a single round *communication optimal* almost-PSMT tolerating threshold adversary. Though there exists single round, communication optimal almost-PSMT protocol tolerating threshold adversary [35], we find that our protocol is much more *computationally efficient* and *relatively simpler* than the protocol of [35]. In practical networks like sensor networks, it is desirable to have protocols which perform simple computation. In such a situation, our communication optimal protocol (tolerating threshold adversary) fits the bill more appropriately than the communication optimal protocol of [35].

In [9] the authors have designed single round almost-PSMT protocol tolerating threshold adversary, which performs simple computations. However, their protocol is *not communication optimal*. On the other hand, our protocol tolerating threshold adversary enjoys the property of *being both simple and also communication optimal*.

In Table 1 and 2, we compare our protocols with the best known almost-PSMT protocols in non-threshold and threshold settings respectively.

**Table 1.** Comparison of our almost-PSMT protocol tolerating $\mathcal{Q}^2$ adversary structure with best known almost-PSMT protocol tolerating $\mathcal{Q}^2$ adversary structure

| Reference | Number of Rounds | Efficient/Inefficient |
|---|---|---|
| [25] | At least three | Inefficient |
| This paper | One | Efficient |

**Table 2.** Comparison of our single round almost-PSMT protocol tolerating threshold adversary with $n = 2t + 1$ with the best known single round almost-PSMT protocol tolerating threshold adversary with $n = 2t + 1$

| Reference | Communication Optimal | Computational Complexity |
|---|---|---|
| [35] | Yes | Efficient (Polynomial in $n$) |
| [9] | **No** | More efficient than [9] |
| This paper | Yes | More efficient than [35] |

### 1.7 Tools and Techniques Used in Our Protocol

To design our protocol, we use *Linear Secret Sharing Scheme* (LSSS) [8]. In addition, we also use a new method of authenticating multiple values concurrently in information theoretic sense. Together this leads to our efficient single round almost-PSMT protocol.

## 2 Primitives

Our protocol involves a negligible error probability of $2^{-\Omega(\kappa)}$. To bound the error probability by $2^{-\Omega(\kappa)}$, our protocol operates over a finite field $\mathbb{F}$, where $|\mathbb{F}| = 2^{\kappa}$. In our protocol, the error probability comes from the fact that adversary has to guess a value (unknown to the adversary), selected uniformly and randomly by **S** from $\mathbb{F}$. If the adversary can correctly guess the value, then the protocol output will be incorrect. However, the probability of this event is $\frac{1}{|\mathbb{F}|} = 2^{-\kappa}$. Without loss of generality, we assume that $\frac{\ell}{|\mathbb{F}|} \approx 2^{-\Omega(\kappa)}$ and hence is negligible (this is assumed in all the previous almost-PSMT protocols). We now discuss LSSS.

### 2.1 Linear Secret Sharing Scheme: LSSS

In a *secret sharing scheme*, a dealer $D$ distributes a secret $s \in \mathbb{F}$, to a set of $n$ parties $\mathcal{P} = \{P_1, \ldots, P_n\}$ in such a way that some subsets of the participants (called as access sets) can reconstruct $s$ from their shares, while the other subsets of the participants (called forbidden sets) have no information about $s$ from their shares. The family of access sets is called an *access structure*. Moreover, we assume that access structure is monotone, which is defined as follows:

**Definition 2.** *An access structure $\Sigma$ is monotone if $A \in \Sigma$ and $A' \supseteq A$, then $A' \in \Sigma$.*

Corresponding to the access structure $\Sigma$, we have the adversary structure $\Gamma = \Sigma^c$, where $c$ denotes the complement. The sets in $\Gamma$ are called as *forbidden* sets. There exists a *computationally unbounded* adversary $\mathcal{A}$, who can control any set in $\Gamma$.

A secret sharing scheme for any monotone access structure $\Sigma$ can be realized by a linear secret sharing scheme (LSSS) [8] as follows: Let $\mathcal{M}$ be a $d \times e$ matrix over $\mathbb{F}$ and $\quad : \{1, \cdots, d\} \to \{1, \cdots, n\}$ be a labeling function, where $d \geq e$ and $d \geq n$.

**Sharing algorithm:**

1. To share a secret $s \in \mathbb{F}$, $D$ first chooses a random vector $\boldsymbol{\rho} \in \mathbb{F}^{e-1}$ and compute a vector
$$\boldsymbol{v} = (v_1, \cdots, v_d)^T = \mathcal{M} \cdot \begin{pmatrix} s \\ \boldsymbol{\rho} \end{pmatrix}. \qquad (1)$$
2. Let
$$\mathsf{LSSS}(s, \boldsymbol{\rho}) = (\mathtt{share}_1, \cdots, \mathtt{share}_n), \qquad (2)$$
where $\mathtt{share}_i = \{v_j \mid \quad (j) = i\}$. The dealer gives $\mathtt{share}_i$ to $P_i$ as a share for $i = 1, \cdots, n$.

**Reconstruction algorithm:** A set of parties $A \in \Sigma$ can reconstruct the secret $s$ if and only if $(1, 0, \cdots, 0)$ is in the linear span of

$$\mathcal{M}_A = \{m_j \mid \quad (j) \in A\},$$

where $m_j$ denotes the $j$th row of $\mathcal{M}$. If this is indeed the case then there exists a vector $\boldsymbol{\alpha}_A$ called *recombination vector*, such that $\boldsymbol{\alpha}_A \cdot \mathcal{M}_A = (1, 0, \ldots, 0)$. Let $\boldsymbol{s}_A$ denote the set of shares corresponding to the parties in $A$. Then the parties in $A$ can reconstruct $s$ by computing $s = \langle \boldsymbol{\alpha}_A, \boldsymbol{s}_A^T \rangle$, where $\langle x, y \rangle$ denotes *dot product* of $x$ and $y$ and $x^T$ denotes transpose of $x$.

**Definition 3 (Monotone Span Programme (MSP) [8]).** *We say that the above $(\mathcal{M}, \ )$ is a monotone span program which realizes $\Sigma$. The size of the MSP is the number of rows $d$ in $M$.*

**Theorem 1 ([8]).** *The above algorithm constitutes a valid secret sharing scheme.*

We are now ready to present our protocol.

## 3 Efficient Single Round Almost-PSMT Protocol Tolerating Non-Threshold Adversary

Let $\mathcal{W} = \{w_1, \ldots, w_n\}$ be the set of $n$ channels between **S** and **R** and let $\mathcal{A}$ be a non-threshold adversary, specified by an adversary structure $\Gamma$ over $\mathcal{W}$. Moreover, let $\Sigma = \Gamma^c$ be the corresponding access structure over $\mathcal{W}$. Furthermore, let $\mathcal{A}$ satisfies $\mathcal{Q}^2$ condition with respect to $\mathcal{W}$, which is necessary for the existence of any almost-PSMT protocol tolerating $\mathcal{A}$. During the protocol, $\mathcal{A}$ can select any set of channels $B \in \Gamma$ for corruption. However, before the beginning of the protocol, neither **S** nor **R** will know which set of channels are under the control of $\mathcal{A}$. The channels which will be under the control of $\mathcal{A}$ will be called *corrupted.* On the other hand, the channels not under the control of $\mathcal{A}$ will be called *honest.*

Let $(\mathcal{M}, \ )$ be the MSP realizing the access structure $\Sigma$. *Without loss of generality and for simplicity, we assume that only $i^{th}$ row of $\mathcal{M}$ is assigned to channel $w_i$, for $i = 1, \ldots, n$.* Thus,

$$\mathcal{M} = \begin{pmatrix} \boldsymbol{m_1} \\ \vdots \\ \boldsymbol{m_n} \end{pmatrix}$$

is an $n \times e$ matrix over $\mathbb{F}$. *However, our protocol will also work when more than one row of $\mathcal{M}$ is assigned to some $w_i$.* Finally we use the following notation in our protocol:

**Notation 1** *Let $\mathcal{Q}$ be any subset of $\mathcal{W}$ i.e $\mathcal{Q} \subseteq \mathcal{W}$. Then $\mathcal{M}_{\mathcal{Q}}$ denotes the matrix containing the rows of $\mathcal{M}$ corresponding to the channels in $\mathcal{Q}$. For example, if $\mathcal{Q} = \{w_1, \ldots, w_t\}$, then*

$$\mathcal{M}_{\mathcal{Q}} = \begin{pmatrix} \boldsymbol{m_1} \\ \vdots \\ \boldsymbol{m_t} \end{pmatrix}.$$

### 3.1 Underlying Idea of the Protocol

The high level idea of the protocol is as follows: let the message $m^{\mathbf{S}}$, which is a sequence of $\ell$ elements from $\mathbb{F}$ be denoted by $m^{\mathbf{S}} = [m_1^{\mathbf{S}}, \ldots, m_\ell^{\mathbf{S}}]$. Now using the MSP $\mathcal{M}$, **S** generates $\mathsf{LSSS}(m_i^{\mathbf{S}}, \boldsymbol{\rho}_i) = (\mathtt{share}_{i1}^{\mathbf{S}}, \ldots, \mathtt{share}_{in}^{\mathbf{S}})$, for $i = 1, \ldots, \ell$, where $\boldsymbol{\rho}_i$'s are the randomness used by **S**.

If $\mathbf{S}$ sends the $j^{th}$ share of all the $\ell$ $m_i^{\mathbf{S}}$'s, namely $\mathtt{share}_{ij}^{\mathbf{S}}$, over $w_j$, for $j = 1, \ldots, n$, then it still preserves the secrecy of $m^{\mathbf{S}}$. This is because $\mathcal{A}$ can control any one set from the adversary structure $\Gamma$ and hence will get the shares of each $m_i^{\mathbf{S}}$'s, sent over those channels. However, from the properties of MSP, these shares will not reveal any information about $m_i^{\mathbf{S}}$'s to $\mathcal{A}$.

However, $\mathbf{S}$ cannot ensure that $m^{\mathbf{S}}$ will be recovered correctly by $\mathbf{R}$ by simply sending the shares. This is because $\mathcal{A}$ may corrupt the shares sent over the channels under its control and there will be no way by which $\mathbf{R}$ can detect which channels have delivered correct shares. This is because there is only one round in the protocol. So $\mathbf{S}$ has also to send some *additional* information to authenticate each share, which can assist $\mathbf{R}$ to detect the corrupted shares with very high probability. So in our protocol, $\mathbf{S}$ also sends additional authentication information, using which $\mathbf{R}$ can detect the corrupted shares with very high probability (the way this is done is explained in the next section).

*Though this mechanism of sending the shares, along with their authentication is used in earlier almost-PSMT protocols, we use a new way of sending the authentication information, which is relatively simpler than the earlier schemes.* After removing the corrupted shares, $\mathbf{R}$ will be left with the shares, which, with very high probability are correctly delivered. Among these shares, there will be a set of shares which are delivered over the channels which are *honest* and hence constitutes an access set. So if $\mathbf{R}$ applies the reconstruction algorithm of the LSSS to the retained shares, $\mathbf{R}$ will correctly recover each $m_i^{\mathbf{S}}$ with very high probability.

### 3.2 Sending the Authentication Information

In our protocol, the authentication of shares is sent in the following way: corresponding to $j^{th}$ share of all the $\ell$ $m_i^{\mathbf{S}}$'s, sender $\mathbf{S}$ constructs a polynomial $p_j^{\mathbf{S}}(x)$ of degree $\ell - 1$ as follows: $p_j^{\mathbf{S}}(x) = \mathtt{share}_{1j}^{\mathbf{S}} + \mathtt{share}_{2j}^{\mathbf{S}} \cdot x + \ldots + \mathtt{share}_{\ell j}^{\mathbf{S}} \cdot x^{\ell-1}$. Now $\mathbf{S}$ associates $p_j^{\mathbf{S}}(x)$ with channel $w_j$, for $j = 1, \ldots, n$ and sends it over $w_j$ (by sending the coefficients of $p_j^{\mathbf{S}}(x)$ over $w_j$). This is the same as sending all the $j^{th}$ shares over $w_j$.

Now $\mathbf{S}$ associates a *random evaluation point* $\alpha_k^{\mathbf{S}}$ with every channel $w_k$, for $k = 1, \ldots, n$. If $\mathbf{S}$ sends $\alpha_k^{\mathbf{S}}$ and $p_j^{\mathbf{S}}(\alpha_k^{\mathbf{S}})$ over $w_k$, then it achieves the following: if $w_j$ is *corrupted* and if $w_k$ is *honest*, then $w_j$ cannot deliver $p_j^{\mathbf{R}}(x) \neq p_j^{\mathbf{S}}(x)$ to $\mathbf{R}$ over $w_j$ without being caught by $w_k$ with very high probability. This is because $\mathcal{A}$ will have no information about $\alpha_k^{\mathbf{S}}$ sent over $w_k$ and also $\alpha_k^{\mathbf{R}}$ received by $\mathbf{R}$ over $w_k$ is same as $\alpha_k^{\mathbf{S}}$. So except with probability $\frac{\ell-1}{|\mathbb{F}|}$, $p_j^{\mathbf{R}}(\alpha_k^{\mathbf{R}}) \neq p_j^{\mathbf{S}}(\alpha_k^{\mathbf{R}})$. This is because two different polynomials of degree $\ell - 1$ can have at most $\ell - 1$ common roots and $\alpha_k^{\mathbf{S}}$ is randomly selected from $\mathbb{F}$. By appropriately selecting $\mathbb{F}$, we can ensure that $\frac{\ell-1}{|\mathbb{F}|} \approx 2^{-\Omega(\kappa)}$, which is negligible. So this can help to detect corrupted shares.

However, the above communication may breach secrecy as follows: if $P_j$ is *honest* and $P_k$ is *corrupted*, then earlier adversary would have no information about $p_j^{\mathbf{S}}(x)$, as no information about $p_j^{\mathbf{S}}(x)$ would have been sent over $w_k$. But now, adversary will know $p^{\mathbf{S}}_j(\alpha^{\mathbf{S}}_k)$, as well as $\alpha^{\mathbf{S}}_k$ through $w_k$, thus revealing information about $p^{\mathbf{S}}_j(x)$ and hence about $j^{th}$ shares of all $m^{\mathbf{S}}_i$'s. To avoid this situation, we use the following idea: corresponding to channel $w_j$, **S** selects $n$ random *masking keys*, denoted by $key^{\mathbf{S}}_{j1}, \ldots, key^{\mathbf{S}}_{jn}$. All the $n$ masking keys (associated with $w_j$) are sent over $w_j$. Now the authentication of $p^{\mathbf{S}}_j(x)$ corresponding to evaluation point $\alpha^{\mathbf{S}}_k$, namely $p^{\mathbf{S}}_j(\alpha^{\mathbf{S}}_k)$ is masked with the $k^{th}$ masking key, namely $key^{\mathbf{S}}_{jk}$ and sent over $w_k$. That is, over $w_k$, **S** sends $p^{\mathbf{S}}_j(\alpha^{\mathbf{S}}_k) + key^{\mathbf{S}}_{jk}$, instead of only $p^{\mathbf{S}}_j(\alpha^{\mathbf{S}}_k)$. Notice that $key^{\mathbf{S}}_{jk}$ is not sent over $w_k$. So if adversary controls $w_k$, then even after knowing $\alpha^{\mathbf{S}}_k$ and $p^{\mathbf{S}}_j(\alpha^{\mathbf{S}}_k) + key^{\mathbf{S}}_{jk}$, adversary will not gain any information about $p^{\mathbf{S}}_j(x)$, as he has no information about the $k^{th}$ masking key $key^{\mathbf{S}}_{jk}$ associated with $w_j$. This way, we preserve the secrecy of each $p^{\mathbf{S}}_j(x)$, sent over honest $p_j$'s. The interesting fact is that with this communication, we can also ensure that if some $p^{\mathbf{S}}_j(x)$ is changed by the adversary over some corrupted $w_j$, then it will be detected with very high probability by an *honest* $w_k$.

We are now ready to formally present our protocol, which is given in Fig. 1.

We now proceed to prove the properties of the protocol. In the proofs, we will use the following notations (For the definition of VALID, see Fig. 1):

- HW denotes the set of channels in $\mathcal{W}$ not under the control of $\mathcal{A}$.
- CW denotes the set of channels in $\mathcal{W}$ under the control of $\mathcal{A}$.
- HVALID denotes the set of channels in VALID not under the control of $\mathcal{A}$.
- CVALID denotes the set of channels in VALID under the control of A.

*Remark 2.* Notice that if some channel is under the control of $\mathcal{A}$ then it is not necessary that $\mathcal{A}$ changes all the information sent over the channel. The adversary may or may not change any portion of the information sent over the channels under his control.

**Lemma 1.** *HVALID = HW and hence HVALID constitutes an access set.*

PROOF: First notice that every channel in the set HW will correctly deliver all the information to **R**. Specifically, $p^{\mathbf{R}}_k(x) = p^{\mathbf{S}}_k(x)$, $\alpha^{\mathbf{R}}_k = \alpha^{\mathbf{S}}_k$, $(key^{\mathbf{R}}_{k1}, \ldots, key^{\mathbf{R}}_{kn}) = (key^{\mathbf{S}}_{k1}, \ldots, key^{\mathbf{S}}_{kn})$ and $val^{\mathbf{R}}_{jk} = val^{\mathbf{S}}_{jk}$, for $j = 1, \ldots, n$, for every channel $w_k \in$ HW. So the condition $val^{\mathbf{R}}_{jk} = p^{\mathbf{R}}_j(\alpha^{\mathbf{R}}_k) + key^{\mathbf{R}}_{jk}$ for every $w_j, w_k \in$ HW. Moreover, HW constitutes an access set. Thus, the condition $\mathcal{W} \setminus \mathsf{SUPPORT}_j \in \Gamma$ will hold for every channel $w_j \in$ HW. Thus, every channel in HW will be present in VALID and hence HVALID = HW. □

**Lemma 2.** *Every channel $w_j \in$ VALID will deliver $p^{\mathbf{R}}_j(x) = p^{\mathbf{S}}_j(x)$, except with error probability $2^{-\Omega(\kappa)}$.*

PROOF: The proof holds without any error probability if $w_j \in$ HVALID. So we now consider the case when $w_j \in$ CVALID. So let $w_j$ be a wire in CVALID. Since $w_j \in$ CVALID (and hence VALID), it implies that $\mathcal{W} \setminus \mathsf{SUPPORT}_j \in \Gamma$. This further implies that there exists at least one channel in $\mathsf{SUPPORT}_j$, say $w_k$, such that $w_k$ is not under the control of the adversary. Otherwise, it implies

**Fig. 1.** Efficient Single Round Almost-PSMT Tolerating $\mathcal{Q}^2$ Adversary Structure

**Computation by S**:

1. For $i = 1, \ldots, \ell$, **S** computes $\mathsf{LSSS}(m_i^{\mathbf{S}}, \boldsymbol{\rho}_i) = (\mathtt{share}_{i1}^{\mathbf{S}}, \ldots, \mathtt{share}_{in}^{\mathbf{S}})$.
2. For $k = 1, \ldots, n$, corresponding to channel $w_k$, **S** selects a random value $\alpha_k^{\mathbf{S}}$, called as $k^{th}$ *evaluation point.*
3. For $j = 1, \ldots, n$, corresponding to the $j^{th}$ share of all the $\ell$ $m_i^{\mathbf{S}}$'s, **S** constructs a polynomial $p_j^{\mathbf{S}}(x)$ of degree $\ell - 1$ as follows:
$$p_j^{\mathbf{S}}(x) = \mathtt{share}_{1j}^{\mathbf{S}} + \mathtt{share}_{2j}^{\mathbf{S}} \cdot x + \ldots + \mathtt{share}_{\ell j}^{\mathbf{S}} \cdot x^{\ell-1}.$$
4. For $j = 1, \ldots, n$, **S** evaluates each $p_j^{\mathbf{S}}(x)$ at evaluation point $\alpha_k^{\mathbf{S}}$, for $k = 1, \ldots, n$.
5. For $j = 1, \ldots, n$, corresponding to channel $w_j$, **S** selects $n$ random, non-zero values $key_{j1}^{\mathbf{S}}, \ldots, key_{jn}^{\mathbf{S}}$, called as *masking keys.*

**Round I: Communication from S to R**: For $k = 1, \ldots, n$, **S** sends the following to **R** over channel $w_k$ and terminates the protocol.

1. Polynomial $p_k^{\mathbf{S}}(x)$.
2. Evaluation point $\alpha_k^{\mathbf{S}}$.
3. $n$ masking keys $key_{k1}^{\mathbf{S}}, \ldots, key_{kn}^{\mathbf{S}}$.
4. Masked authentication values $val_{jk}^{\mathbf{S}}$, for $j = 1, \ldots, n$, where $val_{jk}^{\mathbf{S}} = p_j^{\mathbf{S}}(\alpha_k^{\mathbf{S}}) + key_{jk}^{\mathbf{S}}$.

**Information Received by R**: For $k = 1, \ldots, n$, let **R** receives the following from **S** over channel $w_k$:

1. Polynomial $p_k^{\mathbf{R}}(x)$.
2. Evaluation point $\alpha_k^{\mathbf{R}}$.
3. $n$ masking keys $key_{k1}^{\mathbf{R}}, \ldots, key_{kn}^{\mathbf{R}}$.
4. Masked authentication values $val_{jk}^{\mathbf{R}}$, for $j = 1, \ldots, n$ [a].

**Message Recovery by R**: **R** does the following computation:

1. **R** initializes a set $\mathsf{VALID} = \emptyset$.
2. For $j = 1, \ldots, n$, corresponding to channel $w_j$, **R** constructs a set $\mathsf{SUPPORT}_j = \emptyset$.
3. **R** adds channel $w_k$ in $\mathsf{SUPPORT}_j$ if $val_{jk}^{\mathbf{R}} = p_j^{\mathbf{R}}(\alpha_k^{\mathbf{R}}) + key_{jk}^{\mathbf{R}}$.
4. For $j = 1, \ldots, n$, **R** adds channel $w_j$ to $\mathsf{VALID}$ if $\mathcal{W} \setminus \mathsf{SUPPORT}_j \in \Gamma$.
5. Without loss of generality, let $w_1, \ldots, w_t$ be the channels in $\mathsf{VALID}$. Moreover, for $j = 1, \ldots, t$, let $p_j^{\mathbf{R}}(x)$ be of the form
$$p_j^{\mathbf{R}}(x) = \mathtt{share}_{1j}^{\mathbf{R}} + \mathtt{share}_{2j}^{\mathbf{R}} \cdot x + \ldots + \mathtt{share}_{\ell j}^{\mathbf{R}} \cdot x^{\ell-1}.$$
6. For $i = 1, \ldots, \ell$, **R** applies reconstruction algorithm of the $\mathsf{LSSS}$ to $\mathtt{share}_{i1}^{\mathbf{R}}, \mathtt{share}_{i2}^{\mathbf{R}}, \ldots, \mathtt{share}_{it}^{\mathbf{R}}$ and reconstructs $m_i^{\mathbf{R}}$.
7. Finally **R** reconstructs $m^{\mathbf{R}} = [m_1^{\mathbf{R}}, \ldots, m_\ell^{\mathbf{R}}]$ and terminates the protocol.

[a] If channel $w_k$ is not under the control of $\mathcal{A}$ then $p_k^{\mathbf{R}}(x) = p_k^{\mathbf{S}}(x)$, $\alpha_k^{\mathbf{R}} = \alpha_k^{\mathbf{S}}$, $(key_{k1}^{\mathbf{R}}, \ldots, key_{kn}^{\mathbf{R}}) = (key_{k1}^{\mathbf{S}}, \ldots, key_{kn}^{\mathbf{S}})$ and $val_{jk}^{\mathbf{R}} = val_{jk}^{\mathbf{S}}$, for $j = 1, \ldots, n$.

that $\mathsf{SUPPORT}_j \in \Gamma$ and hence $\mathcal{A}$ does not satisfy $\mathcal{Q}^2$ condition with respect to $\mathcal{W}$, which is a contradiction.

Now since $w_k$ is not under the control of $\mathcal{A}$, it implies that $\alpha_k^{\mathbf{R}} = \alpha_k^{\mathbf{S}}$ and also $val_{jk}^{\mathbf{R}} = val_{jk}^{\mathbf{S}}$. Moreover, $\mathcal{A}$ will have no information about $\alpha_k^{\mathbf{R}}$ and $val_{jk}^{\mathbf{R}}$. Now suppose adversary changes $p_j^{\mathbf{S}}(x)$, so that $p_j^{\mathbf{R}}(x) \neq p_j^{\mathbf{R}}(x)$. However, since $w_k \in \mathsf{SUPPORT}_j$, it implies that $val_{jk}^{\mathbf{R}} = p_j^{\mathbf{R}}(\alpha_k^{\mathbf{R}}) + key_{jk}^{\mathbf{R}}$. But adversary can ensure the same only if he can correctly guess $\alpha_k^{\mathbf{R}} = \alpha_k^{\mathbf{S}}$. However, adversary can do the same with probability at most $\frac{\ell-1}{|\mathbb{F}|} \approx 2^{-\Omega(\kappa)}$. □

**Lemma 3 (Perfect Secrecy).** *The protocol in Fig. 1 satisfies perfect secrecy condition.*

Proof: If $w_k \in \mathsf{CW}$, then adversary will know the polynomial $p_k^{\mathbf{S}}(x)$ and hence the shares $\mathtt{share}_{1k}^{\mathbf{S}}, \ldots, \mathtt{share}_{\ell k}^{\mathbf{S}}$. However, even after knowing all the polynomials transmitted through the channels in $\mathsf{CW}$, adversary will not know $m_1^{\mathbf{S}}, \ldots, m_\ell^{\mathbf{S}}$, as adversary will only come to know the shares of $m_1^{\mathbf{S}}, \ldots, m_\ell^{\mathbf{S}}$ sent through the channels in $\mathsf{CW}$ and $\mathsf{CW} \in \Gamma$. However, the adversary will also know $val_{jk}^{\mathbf{S}} = p_j^{\mathbf{S}}(\alpha_k^{\mathbf{S}}) + key_{jk}^{\mathbf{S}}$, corresponding to every $w_j \in \mathsf{HW}$, which is transmitted through every $w_k \in \mathsf{CW}$. However, such $val_{jk}^{\mathbf{S}}$'s will not reveal any extra information about $p_j^{\mathbf{S}}(x)$ (corresponding to any $P_j$ in $\mathsf{HW}$) to the adversary, as the adversary will have no information about the masking key $key_{jk}^{\mathbf{S}}$, which is only sent over $w_j$. Thus, $val_{jk}^{\mathbf{S}}$'s corresponding to every $w_j \in \mathsf{HW}$, which are transmitted through every $p_k \in \mathsf{CW}$ will not reveal any information about $p_j^{\mathbf{S}}(x)$'s corresponding to $w_j$'s in $\mathsf{HW}$. Thus, through the information received over the channels in $\mathsf{CW}$, adversary will not get any information about $m_i^{\mathbf{S}}$'s and hence the message $m^{\mathbf{S}}$.

**Lemma 4 (Almost Perfect Reliability).** *The protocol in Fig. 1 satisfies almost perfect reliability condition.*

Proof: To prove the lemma, we have to show that the shares (of $m_i^{\mathbf{S}}$'s) received by $\mathbf{R}$ over the channels in $\mathsf{VALID}$ are correct shares, except with error probability $2^{-\Omega(\kappa)}$. This further implies that every channel $w_j \in \mathsf{VALID}$ has delivered $p_j^{\mathbf{R}}(x) = p_j^{\mathbf{S}}(x)$, except with error probability $2^{-\Omega(\kappa)}$. However, this follows from Lemma 2. □

**Lemma 5 (Computation and Communication Complexity).** *In the protocol of Fig. 1,* $\mathbf{S}$ *and* $\mathbf{R}$ *performs computation which is polynomial in the size of* $\Gamma$ *and the underlying* LSSS*. In the protocol,* $\mathbf{S}$ *sends* $\mathcal{O}(\ell n + n^2)$ *field elements from* $\mathbb{F}$ *to* $\mathbf{R}$*.*

Proof: The computation complexity is easy to verify. We now analyze the communication complexity. Through each channel, $\mathbf{S}$ sends a polynomial of degree $\ell - 1$, one evaluation point, $n$ masking keys and $n$ authenticated values. This results in a total communication complexity of $\mathcal{O}(\ell n + n^2)$ field elements. □

**Theorem 2.** *Let* $\mathbf{S}$ *and* $\mathbf{R}$ *be connected by* $n$ *channels and let there exists a computationally unbounded adversary* $\mathcal{A}$*, specified by an adversary structure* $\Gamma$

*over the n channels, such that $\mathcal{A}$ satisfies $\mathcal{Q}^2$ condition. Then there exists an efficient single round almost-PSMT protocol tolerating $\mathcal{A}$.*

PROOF: The proof follows from Lemma 3, Lemma 4 and Lemma 5. □

# 4 Simple and Computationally Efficient Single Round Almost-PSMT Tolerating Threshold Adversary With Optimum Communication Complexity

As discussed earlier, a threshold adversary $\mathcal{A}_t$, with threshold $t$, is a special type of non-threshold adversary where the size of each set in the adversary structure $\Gamma$ is at most $t$. We now recall the following results from [25].

**Theorem 3 ([25]).** *Any almost-PSMT (irrespective of the number of rounds) tolerating $\mathcal{A}_t$ is possible iff* **S** *and* **R** *are connected by $n \geq 2t+1$ channels. Moreover, any single round almost-PSMT protocol tolerating $\mathcal{A}_t$ has to communicate $\Omega\left(\frac{n\ell}{n-2t}\right)$ field elements to send a message containing $\ell$ field elements.*

*Remark 3.* In any almost-PSMT protocol, $|\mathbb{F}|$ is selected as a function of the error parameter $\kappa$ (normally $|\mathbb{F}| = 2^{\kappa}$) and thus each each field element can be represented by a number of bits, which will be function of $\kappa$. So though $\kappa$ is not figuring out explicitly in the expression for communication complexity in Theorem 3, it is implied implicitly if we look into the total number of bits that are actually communicated.

Any single round almost-PSMT protocol designed with $n = 2t + 1$ channels is said to have *optimal resilience*. Substituting $n = 2t+1$ in the above theorem, we find that any single round almost-PSMT protocol with optimal resilience has to communicate $\Omega(n\ell)$ field elements to send a message containing $\ell$ field elements. Thus any single round, optimally resilient, almost-PSMT protocol whose total communication complexity is $\mathcal{O}(n\ell)$ is said to be *communication optimal*.

In [35, 25], the authors presented an efficient[5] single round, optimally resilient almost-PSMT protocol tolerating $\mathcal{A}_t$. However, the protocol performs some complex (though efficient) computational stuffs, like *extrapolation technique, extranding randomness*, etc[6] to achieve its task. In practical networks like sensor network, it is desirable to design protocols which perform computationally simple steps. Motivated by this, the authors in [9] have designed a very simple, optimally resilient, single round almost-PSMT tolerating $\mathcal{A}_t$. However, their protocol is not communication optimal. Specifically, their protocol sends $\mathcal{O}(n^2)$ field elements to send a message containing one field element.

We now show that our single round almost-PSMT protocol against non-threshold adversary when restricted to threshold adversary is a single round,

[5] The computation and communication complexity of the protocol is polynomial in $n$ and $\ell$.

[6] See [7] for the detailed presentation of the single round almost-PSMT protocol of [35].

optimally resilient, almost-PSMT protocol tolerating $\mathcal{A}_t$ having *optimal communication complexity.* Moreover, the protocol is efficient. Furthermore, the protocol is very simple and performs much simplers steps than the communication optimal single round almost-PSMT protocol of [35].

The first observation is that if the adversary is specified by a threshold $t$ and if the underlying adversary structure satisfies $\mathcal{Q}^2$ condition, then it implies that **S** and **R** are connected by $n \geq 2t+1$ channels. Moreover, it is well known that there exists a very simple MSP tolerating a threshold adversary with threshold $t$, such that there are exactly $n$ rows in the MSP and one row of the MSP is assigned to each channel. The MSP is nothing but an $n \times (t+1)$ Vandermonde matrix [8]. The resultant secret sharing scheme is known as Shamir secret sharing scheme [32]. So now with these observations, if we simply execute the protocol of previous section assuming that the adversary is a threshold adversary and there are $n = 2t+1$ channels between **S** and **R**, we get a simple, efficient, optimally resilient, single round almost-PSMT protocol tolerating $\mathcal{A}_t$, which communicates $\mathcal{O}(\ell n + n^2)$ field elements to send a message containing $\ell$ field elements. Now if we set $\ell = n$, then we find that the protocol sends a message containing $n$ field elements by communicating $\mathcal{O}(n^2)$ field elements. From Theorem 3, any single round optimally resilient almost-PSMT protocol has to communicate $\Omega(n^2)$ field elements to securely send a message containing $n$ field elements. Thus our resultant protocol is communication optimal. We now state this in the following theorem:

**Theorem 4.** *Let* **S** *and* **R** *be connected by* $n = 2t+1$ *channels. Moreover, let* **S** *has a message containing* $\ell = n$ *field elements. Then there exists a simple, efficient, optimally resilient, communication optimal single round almost-PSMT protocol tolerating* $\mathcal{A}_t$.

## 5 Conclusion

In this paper, we resolved one of the open problems raised in [25] by designing an optimally resilient, single round, efficient almost-PSMT protocol tolerating non-threshold adversary. This is the first ever efficient single round almost-PSMT protocol tolerating non-threshold adversary. When restricted to threshold adversary, we get a simple, efficient, optimally resilient, single round communication optimal almost-PSMT protocol.

## References


1. 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-Verlag, 2006.

2. T. Araki. Almost secure 1-round message transmission scheme with polynomial time message decryption. In Reihaneh Safavi-Naini, editor, *Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada, August 10-13, 2008, Proceedings*, volume 5155 of *Lecture Notes in Computer Science*, pages 2–13. Springer, 2008.
3. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, 2-4 May 1988, Chicago, Illinois, USA*, pages 1–10. ACM, 1988.
4. D. Chaum, C. Crépeau, and I. Damgård. Multiparty Unconditionally Secure Protocols (extended abstract). In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, 2-4 May 1988, Chicago, Illinois, USA*, pages 11–19. ACM, 1988.
5. A. Choudhary, A. Patra, B. V. Ashwinkumar, K. Srinathan, and C. Pandu Rangan. Perfectly Reliable and Secure Communication Tolerating Static and Mobile Mixed Adversary. In R. Safavi-Naini, editor, *Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada, August 10-13, 2008, Proceedings*, volume 5155 of *Lecture Notes in Computer Science*, pages 137–155. Springer, 2008.
6. A. Choudhary, A. Patra, Ashwinkumar B. V, K. Srinathan, and C. Pandu Rangan. On minimal connectivity requirement for secure message transmission in asynchronous networks. In V. Garg, R. Wattenhofer, and K. Kothapalli, editors, *Distributed Computing and Networking, 10th International Conference, ICDCN 2009, Hyderabad, India, January 03-06, 2009*, volume 5408 of *Lecture Notes in Computer Science*, pages 148–162, 2009.
7. A. Choudhury. Protocols for reliable and secure message transmission. Cryptology ePrint Archive, Report 2010/281, 2010.
8. R. Cramer, I. Damgård, and U. M. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, pages 316–334, 2000.
9. Y. Desmedt, S. Erotokritou, and R. Safavi-Naini. Simple and communication complexity efficient almost secure and perfectly secure message transmission schemes. In D. J. Bernstein and T. Lange, editors, *Progress in Cryptology - AFRICACRYPT 2010, Third International Conference on Cryptology in Africa, Stellenbosch, South Africa, May 3-6, 2010. Proceedings*, volume 6055 of *Lecture Notes in Computer Science*, pages 166–183. Springer, 2010.
10. Y. Desmedt and Y. Wang. Perfectly secure message transmission revisited. In E. Biham, editor, *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 502–517. Springer, 2003.
11. Y. Desmedt, Y. Wang, and M. Burmester. A complete characterization of tolerable adversary structures for secure point-to-point transmissions without feedback. In *Algorithms and Computation, 16th International Symposium, ISAAC 2005, Sanya, Hainan, China, December 19-21, 2005, Proceedings*, volume 3827 of *Lecture Notes in Computer Science*, pages 277–287. Springer Verlag, 2005.
12. D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *JACM*, 40(1):17–47, 1993.
13. M. Fitzi, M. K. Franklin, J. A. Garay, and S. H. Vardhan. Towards optimal and efficient perfectly secure message transmission. In S. P. Vadhan, editor, *Theory*

*of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, volume 4392 of *Lecture Notes in Computer Science*, pages 311–322. Springer, 2007.
14. M. Franklin and R. Wright. Secure communication in minimal connectivity models. *Journal of Cryptology*, 13(1):9–30, 2000.
15. M. Hirt and U. M. Maurer. Complete Characterization of Adversaries Tolerable in Secure Multi-Party Computation. In *Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing, Santa Barbara, California, USA, August 21-24, 1997*, pages 25–34. ACM Press, 1997.
16. M. V. N. A. Kumar, P. R. Goundan, K. Srinathan, and C. Pandu Rangan. On perfectly secure communication over arbitrary networks. In *PODC 2002, Proceedings of the Twenty-First Annual ACM Symposium on Principles of Distributed Computing, July 21-24, 2002 Monterey, California, USA*, pages 193–202. ACM, 2002.
17. K. Kurosawa. General error decodable secret sharing scheme and its application. Cryptology ePrint Archive, Report 2009/263, 2009.
18. K. Kurosawa. Round-efficient perfectly secure message transmission scheme against general adversary. Cryptology ePrint Archive, Report 2010/450, 2010.
19. K. Kurosawa and K. Suzuki. Truly efficient 2 round perfectly secure message transmission scheme. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 324–340. Springer, 2008.
20. K. Kurosawa and K. Suzuki. Almost secure (1-round, $n$-channel) message transmission scheme. *IEICE Transactions*, 92-A(1):105–112, 2009.
21. A. Patra, A. Choudhary, and C. Pandu Rangan. Constant phase efficient protocols for secure message transmission in directed networks. In I. Gupta and R. Wattenhofer, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Principles of Distributed Computing, PODC 2007, Portland, Oregon, USA, August 12-15, 2007*, pages 322–323. ACM, 2007.
22. A. Patra, A. Choudhary, and C. Pandu Rangan. Unconditionally reliable and secure message transmission in directed networks revisited. In R. Ostrovsky, R. De Prisco, and I. Visconti, editors, *Security and Cryptography for Networks, 6th International Conference, SCN 2008, Amalfi, Italy, September 10-12, 2008. Proceedings*, volume 5229 of *Lecture Notes in Computer Science*, pages 309–326. Springer, 2008.
23. A. Patra, A. Choudhary, and C. Pandu Rangan. On communication complexity of secure message transmission in directed networks. In K. Kant, S. V. Premmaraju, K. M. Sivalingam, and J. Wu, editors, *Distributed Computing and Networking, 11th International Conference, ICDCN 2010, Kolkata, India, January 3 - 6, 2010, Proceedings*, volume 5935 of *Lecture Notes in Computer Science*, pages 42–53. Springer Verlag, 2010.
24. A. Patra, A. Choudhary, K. Srinathan, and C. Pandu Rangan. Constant phase bit optimal protocols for perfectly reliable and secure message transmission. In R. Barua and T. Lange, editors, *Progress in Cryptology - INDOCRYPT 2006, 7th International Conference on Cryptology in India, Kolkata, India, December 11-13, 2006, Proceedings*, volume 4329 of *Lecture Notes in Computer Science*, pages 221–235. Springer, 2006.
25. A. Patra, A. Choudhary, K. Srinathan, and C. Pandu Rangan. Unconditionally reliable and secure message transmission in undirected synchronous networks: Possibility, feasibility and optimality. Accepted for publication in International Journal

of Applied Cryptography (IJACT). Also available as Cryptology ePrint Archive: Report 2008/141. A preliminary version appeared in [37], 2009.
26. A. Patra, A. Choudhary, M. Vaidyanathan, and C. Pandu Rangan. Efficient perfectly reliable and secure message transmission tolerating mobile adversary. In Y. Mu, W. Susilo, and J. Seberry, editors, *Information Security and Privacy, 13th Australasian Conference, ACISP 2008, Wollongong, Australia, July 7-9, 2008, Proceedings*, volume 5107 of *Lecture Notes in Computer Science*, pages 170–186. Springer, 2008.
27. A. Patra, A. Choudhury, and C. Pandu Rangan. Brief announcement: perfectly secure message transmissiontolerating mobile mixed adversary with reduced phase complexity. In *PODC*, pages 245–246, 2010.
28. A. Patra, B. Shankar, A. Choudhary, K. Srinathan, and C. Pandu Rangan. Perfectly secure message transmission in directed networks tolerating threshold and non threshold adversary. In F. Bao, S. Ling, T. Okamoto, H. Wang, and C. Xing, editors, *Cryptology and Network Security, 6th International Conference, CANS 2007, Singapore, December 8-10, 2007, Proceedings*, volume 4856 of *Lecture Notes in Computer Science*, pages 80–101. Springer, 2007.
29. T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 73–85. ACM, 1989.
30. H. Sayeed and H. Abu-Amara. Perfectly secure message transmission in asynchronous networks. In *Proceedings of 7th IEEE Symposium on Parallel and Distributed Processing*, pages 100–105. IEEE, 1995.
31. H. Sayeed and H. Abu-Amara. Efficient perfectly secure message transmission in synchronous networks. *Information and Computation*, 126(1):53–61, 1996.
32. A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.
33. P.W. Shor. Polynomial time algorithms for Prime factorization and Discrete Logarithms on a Quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.
34. K. Srinathan. Secure distributed communication. PhD Thesis, IIT Madras, 2006.
35. K. Srinathan, A. Choudhary, A. Patra, and C. Pandu Rangan. Efficient Single Phase Unconditionally Secure Message Transmission with Optimum Communication Complexity. In R. A. Bazzi and B. Patt-Shamir, editors, *Proceedings of the Twenty-Seventh Annual ACM Symposium on Principles of Distributed Computing, PODC 2008, Toronto, Canada, August 18-21, 2008*, page 457. ACM, 2008.
36. K. Srinathan, A. Narayanan, and C. Pandu Rangan. Optimal perfectly secure message transmission. In M. K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 545–561. Springer, 2004.
37. K. Srinathan, A. Patra, A. Choudhary, and C. Pandu Rangan. Probabilistic perfectly reliable and secure message transmission - possibility, feasibility and optimality. In K. Srinathan, C. Pandu Rangan, and M. Yung, editors, *Progress in Cryptology - INDOCRYPT 2007, 8th International Conference on Cryptology in India, Chennai, India, December 9-13, 2007, Proceedings*, volume 4859 of *Lecture Notes in Computer Science*, pages 101–122. Springer, 2007.
38. K. Srinathan, N. R. Prasad, and C. Pandu Rangan. On the optimal communication complexity of multiphase protocols for perfect communication. In *2007 IEEE Symposium on Security and Privacy (S&P 2007), 20-23 May 2007, Oakland, California, USA*, pages 311–320. IEEE Computer Society, 2007.

39. K. Srinathan, P. Raghavendra, and C. Pandu Rangan. On proactive perfectly secure message transmission. In J. Pieprzyk, H. Ghodosi, and E. Dawson, editors, *Information Security and Privacy, 12th Australasian Conference, ACISP 2007, Townsville, Australia, July 2-4, 2007, Proceedings*, volume 4586 of *Lecture Notes in Computer Science*, pages 461–473. Springer, 2007.
40. Q. Yang and Y. Desmedt. Cryptanalysis of secure message transmission protocols with feedback. In *ICITS*, volume 5973 of *Lecture Notes in Computer Science*, pages 159–176, 2009.
41. Q. Yang and Y. Desmedt. General perfectly secure message transmission using linear codes. In *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 448–465, 2010.