Secure Message Transmission in Asynchronous Networks[☆]

Ashish Choudhury[1,6], Arpita Patra[*,2,6], Ashwinkumar B. V[3], Kannan Srinathan[4], C. Pandu Rangan[5]

Abstract

In the *Perfectly Secure Message Transmission* (PSMT) problem, a sender **S** and a receiver **R** are part of a distributed network and connected through n node disjoint paths, also called as *wires*, among which at most t wires are controlled by a static, Byzantine adversary $\mathcal{A}_t^{static}$, having *unbounded computing power*. **S** has a message m, which **S** intends to send to **R**. The challenge is to design a protocol, such that at the end of the protocol, **R** should correctly output m without any error (perfect reliability) and $\mathcal{A}_t^{static}$ should not get *any* information about m, what so ever, in information theoretic sense (perfect security). The problem of *Statistically Secure Message Transmission* (SSMT) is same as PSMT, except that **R** should correctly output m with very high probability. Sayeed et al. [37] have given a PSMT protocol in an asynchronous network tolerating $\mathcal{A}_t^{static}$, where **S** and **R** are connected by $n = 2t + 1$ wires. However, we show that their protocol does not provide perfect security. We then prove that in an asynchronous network, if all the n wires are directed from **S** to **R**, then any PSMT protocol tolerating $\mathcal{A}_t^{static}$ is possible iff $n > 3t$. Surprisingly, we also

[☆] A preliminary Version of this paper appeared in [13]

[*] Corresponding author

Email addresses: partho_31@yahoo.co.in, partho31@gmail.com (Ashish Choudhury), arpitapatra10@gmail.com, arpita@cs.au.dk, arpitapatra_10@yahoo.co.in (Arpita Patra), ashwinkumarbv@gmail.com (Ashwinkumar B. V), srinathan@iiit.ac.in (Kannan Srinathan), prangan55@gmail.com, prangan55@yahoo.com (C. Pandu Rangan)

[1] Applied Statistics Unit, 203 B. T. Road, Indian Statistical Institute Kolkata India 700108.

[2] Ada 223, Department of Computer Science, Aarhus University, Abogade 34, 8200 Arhus, Denmark.

[3] Department of Computer Science and Engineering, Cornell University, U. S. A. The work was done when the author was an undergraduate student at IIT Madras.

[4] Center for Security, Theory and Algorithmic Research, IIIT Hyderabad, India 500032. The work was done when the author was visiting IIT Madras.

[5] Department of Computer Science and Engineering, IIT Madras, Chennai India 600036.
Work Supported by Project No. CSE/05/06/076/DITX/CPAN on Protocols for Secure Computation and Communication, Sponsored by Department of Information Technology, Govt. of India.

[6] The work was done when the first two authors were PhD students at Department of Computer Science, IIT Madras. The first author was supported by Infosys PhD fellowship during his PhD. The second author was supported by Microsoft Research India PhD fellowship during her PhD.

prove that even if all the n wires are bi-directional, then any PSMT protocol in asynchronous network tolerating $\mathcal{A}_t^{static}$ is possible iff $n > 3t$. This is quite surprising because for synchronous networks, by the results of Dolev et al. [16], if all the wires are unidirectional (directed from **S** to **R**), then PSMT tolerating $\mathcal{A}_t^{static}$ is possible iff $n > 3t$, where as if all the wires are bi-directional then PSMT tolerating $\mathcal{A}_t^{static}$ is possible iff $n > 2t$. This shows that *asynchrony of the network demands higher connectivity of the network for the existence of PSMT protocols*. Interestingly, we further show that $n > 2t$ wires are necessary and sufficient for the existence of any SSMT protocol in asynchronous network tolerating $\mathcal{A}_t^{static}$, irrespective of whether the n wires are unidirectional from **S** to **R** or the n wires are bi-directional. By the results of [18, 23], $n > 2t$ are necessary and sufficient for the existence of SSMT in synchronous networks, irrespective of whether the n wires are unidirectional from **S** to **R** or the n wires are bi-directional. This shows that *asynchrony of the network does not demand higher connectivity of the network for SSMT protocols*.

Key words: Information Theoretic Security, Error Probability, Optimality, Asynchronous Networks.

---

1. Introduction

Consider the following problem: there exists a sender **S** and a receiver **R**, who are part of an unreliable, distributed network and connected through n vertex disjoint paths/channels called *wires*. Moreover, they do not share any information beforehand. The distrust in the network is modeled by a centralized entity called *adversary*. The adversary, denoted as $\mathcal{A}_t^{static}$, is a static adversary, having *unbounded computing power*, who can corrupt t out of n wires in Byzantine fashion [7]. **S** has a message m, chosen from a finite field $\mathbb{F}$, which he wants to send to **R**. The goal is to design a protocol, such that at the end of the protocol, **R** should correctly output m without any error. This problem is called as *perfectly reliable message transmission* (PRMT). The problem of *perfectly secure message transmission* (PSMT) has an additional requirement that at the of the protocol, the adversary should get no information about m.

In PRMT and PSMT, the protocols guarantee the delivery of the message without any error. If a negligible error probability is allowed in the message delivery, then we arrive at the notion of *statistically reliable message transmission* (SRMT) and *statistically secure message transmission* (SSMT) respectively. The problem of SRMT [18] is same as PRMT, except that **R** should correctly output m with probability at least $1 - 2^{-\Omega(\kappa)}$, where κ is a given error parameter. Similarly, the problem of SSMT [18] is same as SRMT, with an additional requirement that the adversary should not get any information about m in information theoretic sense.

[7] If a wire is Byzantine corrupted then the adversary has full control over the wire and hence the adversary can force the wire to behave in any arbitrary manner.

PRMT, PSMT, SRMT and SSMT are fundamental problems in secure distributed computing. If **S** and **R** are directly connected by a secure link, as assumed in many fault tolerant distributed computing protocols like secure multiparty computation (MPC) [8, 20, 36, 44, 3, 10, 21, 4, 5, 6], Byzantine agreement (BA) [34, 15, 24, 11, 1, 31] verifiable secret sharing (VSS) [12, 16, 8, 19, 27], then reliable and secure communication between **S** and **R** is trivial. However, it is impractical to assume the existence of a direct link between every two nodes in the network. In such a situation PRMT/PSMT/SRMT/SSMT protocols help to simulate a *virtual* reliable/secure link between **S** and **R**. Thus using these protocols, we can simulate a virtual complete network and then we can execute the above fault tolerant distributed computing protocols.

1.1. Existing Literature

PRMT and PSMT problem was first introduced and studied by Dolev et al. [16]. Assuming the underlying network to be undirected and synchronous, Dolev et al. abstracted the underlying network and assumed that **S** and **R** are connected by n bi-directional vertex disjoint paths, also called as *wires*, of which at most t could be under the control of $\mathcal{A}_t^{static}$ [8]. In such a model, any protocol is assumed to be executed in phases, where a phase is a send from **S** to **R** or vice-versa. So in a single phase protocol, only **S** is allowed to communicate to **R**, while in a multi phase protocol, both **S** and **R** are allowed to communicate with each other along the n wires. Hence while in a single phase protocol, the n wires can be viewed as unidirectional, directed from **S** to **R**, in a multi phase protocol, they can be viewed as bi-directional. Dolev et al. have given the necessary and sufficient condition on the connectivity requirement (number of wires n) for the existence of single and multi phase PSMT protocols, as shown in Table 1. More recent efforts using the same adversarial model for the problem of PSMT include [38, 42, 2, 17, 30, 22].

SRMT and SSMT problem was introduced by Franklin et al. [18] and later studied by [14, 23, 40, 32] in synchronous network. The necessary and sufficient condition on the connectivity requirement (number of wires n) for the existence of single and multi phase SSMT, tolerating $\mathcal{A}_t^{static}$ is given in Table 1.

1.2. Our Motivation and Contribution

The existing results for PSMT and SSMT assumes the underlying network to be synchronous. Thus, if **S** (**R**) sends some information along a wire, then it is assumed that **R** (**S**) will get the information (possibly corrupted, if the wire is under the control of the adversary) along the wire after a fixed interval of time. However, this is a very strong assumption because the delay in the arrival of a single message will affect the overall security of the protocol. A typical large network like the Internet can be modeled more accurately by

[8] The approach of abstracting the network as a collection of n wires is justified using Menger's theorem [26] which states that a graph is $c - (\mathbf{S}, \mathbf{R})$-connected iff **S** and **R** are connected by at least c vertex disjoint paths.

Table 1: Connectivity requirement for single and multi phase PSMT and SSMT protocols in synchronous networks tolerating $\mathcal{A}_t^{static}$.

PSMT		SSMT	
number of phases	number of wires (n)	number of phases	number of wires (n)
1	$n \geq 3t+1$ [16]	1	$n \geq 2t+1$ [14, 23]
≥ 2	$n \geq 2t+1$ [16]	≥ 2	$n \geq 2t+1$ [18]

asynchronous networks than synchronous networks. The inherent difficulty in designing a protocol in asynchronous network comes from the fact that we cannot distinguish between a slow sender and a corrupted sender. Thus a receiver cannot wait to receive message along all the n wires, as waiting for all of them may turn out to be endless. So the receiver has to start computation as soon as he receives information along $n-t$ wires. As a result of this, information along t (potentially honest) wires may get neglected.

In the literature, very little attention has been paid to the study of PSMT and SSMT protocols in asynchronous network due to its complexity. This motivates us to study PSMT and SSMT protocols in asynchronous networks. Our contributions in this paper are as follows:

1. In [37], Sayeed et al. have given a PSMT protocol in asynchronous network, tolerating $\mathcal{A}_t^{static}$ in the presence of $n = 2t+1$ unidirectional wires from $\mathbf{S}$ to $\mathbf{R}$. However, we show that their protocol does not provide perfect security.
2. We show that if there are n *unidirectional* wires from $\mathbf{S}$ to $\mathbf{R}$ in an asynchronous network, then there exists a PSMT protocol tolerating $\mathcal{A}_t^{static}$, iff $n > 3t$. Comparing this with first row of the Table 1, we find that asynchrony of the network does *not* effect the possibility of PSMT protocol, if all the n wires are unidirectional from $\mathbf{S}$ to $\mathbf{R}$.
3. We show that if there are n *bi-directional* wires between $\mathbf{S}$ and $\mathbf{R}$ in an asynchronous network, then there exists a PSMT protocol tolerating $\mathcal{A}_t^{static}$, iff $n > 3t$. This is surprising because from second row of the Table 1, $n > 2t$ bi-directional wires are necessary and sufficient for the existence of PSMT protocol against $\mathcal{A}_t^{static}$ in synchronous network. This shows that if all the n wires are bi-directional, then asynchrony of the network significantly affects the possibility of PSMT protocols.
4. We show that SSMT between $\mathbf{S}$ and $\mathbf{R}$ is possible in an asynchronous network, tolerating $\mathcal{A}_t^{static}$ iff $n > 2t$. Moreover, this is true, irrespective of whether the n wires are unidirectional or the n wires are bi-directional. Comparing this with the results in the Table 1, we find that irrespective of whether the n wires are unidirectional or bi-directional, asynchrony of the network does *not* affect the possibility of SSMT.

In [41], the authors have studied PSMT and SSMT problem in asynchronous networks tolerating a generalized non-threshold adversary, specified by an ad-

versary structure. Informally, an adversary structure specifies the collection of potential corruptible sets of wires, where any one of the sets will be activated/corrupted during the protocol execution. Moreover, the size of each set in the adversary structure need not be same and they can be different. So, essentially the adversary can choose any one of the sets from the adversary structure and can corrupt the wires in the set during protocol execution. It is easy to see that $\mathcal{A}_t^{static}$ is a special type of non-threshold adversary, where size of each set in the adversary structure is at most t. In [41], the authors have given the necessary and sufficient conditions for PSMT and SSMT in asynchronous networks tolerating non-threshold adversary. However, though not explicitly stated in the paper, their characterization for PSMT is true under the assumption that **S** is honest, while their characterization for SSMT is true under the assumption that **S** may be corrupted, where if **S** is corrupted, then he may not send anything to **R** along some path. Note that while in synchronous model, **S** being honest or dishonest does not make any sense, in asynchronous model this makes lots of difference. This is because we cannot distinguish between a slow sender and a corrupted sender. However, in this paper we derive all the necessary and sufficient condition, assuming **S** to be honest. The protocols given in [41] against non-threshold adversary are very complex. Though we can derive protocols for tolerating $\mathcal{A}_t^{static}$ from the protocols of [41] tolerating non-threshold adversary, the resultant protocols will be very complex and inefficient. Instead, since we work on threshold model (where the corruption capability of the adversary is bounded by a threshold), our protocols are very elegant and efficient.

Asynchronous PSMT/SSMT is an important primitive for perfectly/statistically secure multiparty computation over asynchronous incomplete networks. Thus, our results can be used to transform the asynchronous secure computation protocols that run over a complete network [9, 7, 43, 35, 5, 29, 31, 28, 33] into ones that can be executed over incomplete networks.

1.3. Organization of the Paper

In the next section, we discuss the network model and present the definitions. In Section 3 we perform the security analysis of the APSMT protocol of [37] and show that it does not provide perfect security. This is followed by the true characterization of APSMT protocols in the presence of unidirectional channels in Section 4. Section 5 provides characterization of APSMT protocols in the presence of bidirectional channels. Characterization of ASSMT protocols in the presence of unidirectional channels and bidirectional channels are presented in Section 6 and Section 7 respectively. We conclude the paper in Section 8.

2. Model and Definitions

We consider a completely asynchronous network $\mathcal{N}$, where **S, R** are two special nodes in $\mathcal{N}$. All the nodes in $\mathcal{N}$ are modeled as probabilistic interactive Turing Machines, where randomization is achieved through random coins. The corruption in the network is modeled by a *centralized adversary* $\mathcal{A}_t^{static}$, who

has *unbounded computing power* and can actively control at most t nodes in the network, excluding **S** and **R** in Byzantine fashion. $\mathcal{A}_t^{static}$ actively corrupting a node implies that it takes full control of the node and forces the node to (mis)behave in an arbitrary manner [9]. The adversary is *centralized* and *static* who corrupts at most t nodes at the beginning of the execution of the protocol. However, neither **S** nor **R** will know the identity of the nodes under the control of the adversary. A node under the control of $\mathcal{A}_t^{static}$ will remain under its control throughout the protocol.

Following the approach of Dolev et al. [16], we abstract the network and assume that **S** and **R** are connected by n vertex disjoint paths, called *wires*, of which at most t could be actively controlled by $\mathcal{A}_t^{static}$ in Byzantine fashion. Moreover, we consider the following two extreme cases:

1. When all the n wires are directed from **S** to **R**, thus do not allowing any interaction between **S** and **R**;
2. When all the n wires are bi-directional, thus allowing interaction between **S** and **R**.

A wire which is not under the control of $\mathcal{A}_t^{static}$ is called *honest*. To model the asynchrony in the network, we assume that the adversary can schedule the message delivery along every wire; i.e., he can determine the time delay of all the messages along all the n wires. However, adversary can only schedule the messages sent along honest wires, without having any access to them. Moreover, the message sent over an honest wire will be eventually delivered. If a wire is under the control of $\mathcal{A}_t^{static}$, then $\mathcal{A}_t^{static}$ may indefinitely block the communication along the wire. So the receiver may have to wait indefinitely for the message(s) along that wire. Hence the receiver can not distinguish between honest wires which are slow (due to the malicious scheduling of messages by $\mathcal{A}_t^{static}$ on these wires) and corrupted wires which withhold/does not send information at all.

In our protocols, **S** and **R** does computation over a field $\mathbb{F}$, where $\mathbb{F}$ is a finite field of prime order. For PSMT protocols, the only restriction on $\mathbb{F}$ is that $|\mathbb{F}| > n$. On the other hand, for SSMT protocol, $\mathbb{F} = GF(2^\kappa)$, where κ is the error parameter of the protocol. If some $x \in \mathbb{F}$ is sent through all the wires, then it is said to be broadcasted. If x is broadcasted over at least $2t+1$ wires, then receiver will always correctly recover it. This is because out of the $2t+1$ wires, at least $t+1$ will be honest and will eventually deliver x. So the receiver can wait for a value which is received identically over at least $t+1$ wires. We now define *asynchronous perfectly secure message transmission* (APSMT) and *asynchronous statistically secure message transmission* (ASSMT).

Let the message to be transmitted securely be drawn from $\mathbb{F}$ and Γ denote the underlying probability distribution on $\mathbb{F}$. We define the View of a node $P_j \in \mathcal{N}$, at any point of the execution of a protocol Π for secure message transmission,

[9] This subsumes fail-stop corruption, where a corrupted node does not respond at all. In addition, it also subsumes passive corruption, where a corrupted node correctly follows the protocol, but the adversary eavesdrop the computation and communication of the node.

to be the information that P_j can get from its local input to the protocol (if any), all the messages that P_j had earlier sent or received, the protocol code executed by P_j and random coins of P_j. The View of $\mathcal{A}_t^{static}$ at any point of the execution of Π is defined as all the information that $\mathcal{A}_t^{static}$ can get from the Views of all the nodes corrupted by $\mathcal{A}_t^{static}$ (i.e. all the information that these nodes can commonly compute from their Views). For message $m \in \mathbb{F}$, any t-active threshold adversary characterized by $\mathcal{A}_t^{static}$ and any protocol Π for secure message transmission, let $\widehat{\Gamma}(\mathcal{A}_t^{static}, m, \Pi)$ denote the probability distribution on the View of the adversary $\mathcal{A}_t^{static}$ at the end of the execution of Π.

Definition 1 (APSMT). *A protocol Π is said to facilitate asynchronous perfectly secure message transmission (APSMT) between $\mathbf{S}$ and $\mathbf{R}$ if for any message m drawn from $\mathbb{F}$ and for every adversary $\mathcal{A}_t^{static}$, the following conditions are satisfied:*

1. *Perfect Secrecy: $\widehat{\Gamma}(\mathcal{A}_t^{static}, m, \Pi) = \widehat{\Gamma}(\mathcal{A}_t^{static}, m', \Pi)$. That is, the two distributions are identical irrespective of the message transmitted.*
2. *Perfect Reliability: $\mathbf{R}$ should receive m correctly, without any error.*
3. *Termination: $\mathbf{R}$ should eventually terminate the protocol.*

Definition 2 (ASSMT). *A protocol Π is said to facilitate asynchronous statistically secure message transmission (ASSMT) between $\mathbf{S}$ and $\mathbf{R}$ if a negligible error probability of $2^{-\Omega(\kappa)}$ can be tolerated with respect to the Perfect Reliability condition of APSMT by Π, where κ is the error parameter. That is, $\mathbf{R}$ should correctly receive m with probability at least $(1 - 2^{-\Omega(\kappa)})$. The probability is over the choice of m and the coin flips of all the nodes in $\mathcal{N}$ and $\mathcal{A}_t^{static}$.*

3. Security Analysis of APSMT Protocol of [37]

In [37], the authors have given an APSMT protocol tolerating $\mathcal{A}_t^{static}$, where $\mathbf{S}$ and $\mathbf{R}$ are connected by wires w_i, for $i = 1, \ldots, n$, directed from $\mathbf{S}$ to $\mathbf{R}$, where $n = 2t + 1$. We briefly recall the protocol from [37] and show that the protocol does not achieve perfect secrecy; i.e., $\mathcal{A}_t^{static}$ can recover m. In the protocol, message m belongs to the set $Q = \{1, 2, \ldots, m_{max}\}$ of positive integers, such that $m_{max} > n$. Let $MAX = 2m_{max} + 1$. $\mathbf{S}$ sends m by doing the following computation and communication:

1. $\mathbf{S}$ randomly selects n values $K_1, K_2, \ldots, K_n$ from the set Q and associates K_i with wire w_i. For each K_i, $\mathbf{S}$ forms a *key_carrying_polynomial* $p_i(x)$ of degree t, where $p_i(0) = K_i$ and other coefficients of $p_i(x)$ are randomly chosen from Q. $\mathbf{S}$ also forms a *secret_carrying_polynomial* $M(x)$ of degree n, where $M(0) = m$ and the coefficient of x^i is K_i.
2. Through wire w_i, $\mathbf{S}$ sends to $\mathbf{R}$ the value $p_j(i)$, for $j = 1, \ldots, n$. $\mathbf{S}$ also broadcasts $M(1)$ and $M(MAX)$, where the values of $M(x)$ are in N, the infinite set of positive integers.

We now show how $\mathcal{A}_t^{static}$ can recover m from the values sent by **S**. In the protocol, **S** broadcasts:

$$\begin{aligned} V_1 &= M(1) = m + K_1 + K_2 + \ldots + K_n \text{ and} \\ V_2 &= M(MAX) = m + K_1 * MAX + K_2 * MAX^2 + \ldots + K_n * MAX^n \end{aligned}$$

Note that V_1 and V_2 does not belong to Q. They belong to N, the infinite set of positive integers; i.e., the protocol works with the exact values of V_1, V_2. However, $m \in Q$ and is always less than MAX. Since, V_1 and V_2 are broadcasted, $\mathcal{A}_t^{static}$ will also know V_1 and V_2. Also MAX is a publicly known parameter. If $\mathcal{A}_t^{static}$ computes (V_2 mod MAX), then he obtains m, because all other terms in V_2 are multiple of MAX, except m, which is less than MAX. Thus, protocol of [37] does not provide perfect secrecy. In fact, there does not exist any APSMT protocol tolerating $\mathcal{A}_t^{static}$ with $n = 2t + 1$ unidirectional wires from **S** to **R**. In the sequel, we present the true characterization of APSMT protocol tolerating $\mathcal{A}_t^{static}$, when all the n wires are unidirectional from **S** to **R**.

4. APSMT When All Wires are Unidirectional from S to R

In the last section, we saw that the APSMT protocol of [37] does not provide perfect security. We now give the true characterization for APSMT protocols tolerating $\mathcal{A}_t^{static}$, when all the n wires are unidirectional, directed from **S** to **R**.

Theorem 1. *Suppose there exists n wires, directed from* **S** *to* **R**, *of which at most t could be under the control of $\mathcal{A}_t^{static}$. Then there exists an APSMT protocol only if $n > 3t$.*

PROOF: From [16], we know that $n > 3t$ wires are necessary for the existence of any synchronous PSMT protocol tolerating a t-active Byzantine adversary, when all the wires are unidirectional from **S** to **R**. Hence it is obviously necessary for the existence of APSMT protocol tolerating $\mathcal{A}_t^{static}$, if all the wires are unidirectional from **S** to **R**. □

We now show that $n > 3t$ unidirectional wires from **S** to **R** are also sufficient for designing an APSMT protocol. Before that we briefly describe the properties of Reed-Solomon codes [25], which are used in our protocol.

4.1. Reed-Solomon (RS) Codes

Let $\mathbb{F}$ be a finite field and $\alpha_1, \alpha_2, \ldots \alpha_n$ be distinct elements of $\mathbb{F}$. Given $k < n \leq |\mathbb{F}|$, and an arbitrary message block $\mathbf{B} = [m_1\ m_2\ \ldots\ m_k]$, the encoding function for the Reed-Solomon code is defined as $[p_{\mathbf{B}}(\alpha_1)\ p_{\mathbf{B}}(\alpha_2)\ \ldots\ p_{\mathbf{B}}(\alpha_n)]$ where $p_{\mathbf{B}}(x)$ is the polynomial $\sum_{i=0}^{k-1} m_{i+1} x^i$.

Let $W = \{(i_1, a_1), (i_2, a_2), \ldots, (i_l, a_l)\}$ be an input word which differs from a valid RS codeword, say C, at most at r locations. Moreover, let C be the RS codeword corresponding to a message block of size $k + 1$. Then there exists

efficient error correcting procedure, like Berlekamp-Welch algorithm [25], that can correct r errors in W, provided that $|W| \geq k + 2r + 1$ [25]. We denote such an error correcting procedure as $RS - DEC(k, r, W)$, which takes as input a word W and tries to output a polynomial of degree k by correcting at most r errors in W. We are now ready to present our APSMT protocol, which is given in the next section.

4.2. APSMT Protocol in the Presence of $n = 3t + 1$ Unidirectional Wires

Let $\mathbf{S}$ and $\mathbf{R}$ be connected by unidirectional wires $w_i, 1 \leq i \leq n$, which are directed from $\mathbf{S}$ to $\mathbf{R}$, where $n = 3t + 1$. We design an APSMT protocol called $\Pi_{APSMT}^{Unidirectional}$, tolerating $\mathcal{A}_t^{static}$. The protocol is given in Fig. 1.

Figure 1: Protocol $\Pi_{APSMT}^{Unidirectional}$ with $n = 3t + 1$ unidirectional wires from $\mathbf{S}$ to $\mathbf{R}$.

Computation and Communication by S:

1. $\mathbf{S}$ selects a random polynomial $p(x)$ of degree t over $\mathbb{F}$, such that $p(0) = m$, where m is the secret message, which $\mathbf{S}$ wants to send to $\mathbf{R}$.
2. For $i = 1, \ldots, n$, $\mathbf{S}$ sends the tuple $(i, p(i))$ to $\mathbf{R}$ over wire w_i.

Message Recovery by R:

For $r = 0, \ldots, t$, $\mathbf{R}$ does the following in iteration r:

1. Let $\mathcal{W}$ denote the set of wires over which $\mathbf{R}$ has received the tuples and I_r denote the tuples received over the wires in $\mathcal{W}$, when $\mathcal{W}$ contains $2t + 1 + r$ wires.
2. Wait until $|\mathcal{W}| \geq 2t + 1 + r$. $\mathbf{R}$ applies $RS - DEC(t, r, I_r)$ to get the polynomial $p'(.)$. If no polynomial is output, then $\mathbf{R}$ skips the next step and proceeds to next iteration.
3. If for at least $2t + 1$ elements $(i, a) \in I_r$, $p'(i) = a$, then $\mathbf{R}$ outputs $p'(0)$ as the secret message and terminates. Otherwise, $\mathbf{R}$ proceeds to the next iteration.

We now prove the properties of protocol $\Pi_{APSMT}^{Unidirectional}$.

Theorem 2. *In protocol $\Pi_{APSMT}^{Unidirectional}$, the adversary $\mathcal{A}_t^{static}$ gets no information about the secret message m.*

PROOF: It is easy to see that $\mathcal{A}_t^{static}$ gets at most t distinct points on $p(x)$. So $\mathcal{A}_t^{static}$ lacks by one point to uniquely interpolate $p(x)$. This implies that $p(0) = s$ is information theoretically secure. □

Theorem 3. *In protocol $\Pi_{APSMT}^{Unidirectional}$, $\mathbf{R}$ will eventually output m.*

PROOF: Suppose $\mathcal{A}_t^{static}$ corrupts $\hat{r} \leq t$ wires during the transmission of values of $p(x)$. Now during $\hat{r}^{th}$ iteration, $\mathbf{R}$ receives $2t+1+\hat{r}$ points on $p(x)$, of which $\hat{r}$ are corrupted. So from the properties of Reed-Solomon codes [25] (as described in the previous section), polynomial $p'(.)$ which is output by $RS-DEC$ during $\hat{r}^{th}$ iteration will pass through at least $2t+1$ points in I_r. Since out of these $2t+1$ points, at least $t+1$ are honest and uniquely define the original polynomial $p(.)$ ($t+1$ points uniquely define a t degree polynomial), the output polynomial $p'(.)$ is same as $p(.)$. Thus $p(.)$ will be output in $\hat{r}^{th}$ iteration and all the iterations up to iteration $\hat{r}$ will be unsuccessful, as either they will not output any t degree polynomial or the output polynomial will not pass through $2t+1$ points in I_r. □

Theorem 4. *Let there exists n unidirectional wires from $\mathbf{S}$ to $\mathbf{R}$. Then APSMT tolerating $\mathcal{A}_t^{static}$ is possible iff $n > 3t$.*

PROOF: The proof follows from Theorem 1 and protocol $\Pi_{APSMT}^{Unidirectional}$. □

5. APSMT When All Wires are Bidirectional Between S and R

In this section, we characterize APSMT tolerating $\mathcal{A}_t^{static}$, when all the n wires between $\mathbf{S}$ and $\mathbf{R}$ are bi-directional. In this setting, we show that APSMT tolerating $\mathcal{A}_t^{static}$ is possible iff there exists $n > 3t$ bi-directional wires between $\mathbf{S}$ and $\mathbf{R}$. This shows that irrespective of whether the n wires between $\mathbf{S}$ and $\mathbf{R}$ are uni-directional or bi-directional, $n > 3t$ wires are necessary for the existence of any APSMT protocol tolerating $\mathcal{A}_t^{static}$.

Theorem 5. *Let $\mathbf{S}$ and $\mathbf{R}$ be connected by $n = 3t+1$ bi-directional wires, of which at most t are under the control of $\mathcal{A}_t^{static}$. Then there exists an APSMT protocol tolerating $\mathcal{A}_t^{static}$.*

PROOF: Any bi-directional wire between $\mathbf{S}$ and $\mathbf{R}$ can be treated as an unidirectional wire from $\mathbf{S}$ to $\mathbf{R}$. Now we know that there exists an APSMT protocol $\Pi_{APSMT}^{Unidirectional}$ tolerating $\mathcal{A}_t^{static}$ if there exists $n = 3t+1$ unidirectional wires from $\mathbf{S}$ to $\mathbf{R}$. Hence the same protocol can also be executed if there exists $n = 3t+1$ bi-directional wires between $\mathbf{S}$ and $\mathbf{R}$. □

We now show that if all the n wires between $\mathbf{S}$ and $\mathbf{R}$ are bi-directional, then APSMT tolerating $\mathcal{A}_t^{static}$ is possible only if $n > 3t$. The proof is by contradiction. We first show that there does not exist any APSMT protocol between a sender $\mathbf{S}'$ and receiver $\mathbf{R}'$, with three bi-directional wires between $\mathbf{S}'$ and $\mathbf{R}'$, of which one can be corrupted by the adversary (Theorem 6). Then by using a standard player partitioning argument [24], we show that if there exists an APSMT protocol tolerating $\mathcal{A}_t^{static}$ with $n = 3t$ bi-directional wires between $\mathbf{S}$ and $\mathbf{R}$, then there exists an APSMT protocol between $\mathbf{S}'$ and $\mathbf{R}'$ who are connected by three bi-directional wires, of which at most one could be corrupted, which is a contradiction (Theorem 7).

Theorem 6. *Let there exists three bi-directional wires between a sender* $\mathbf{S}'$ *and a receiver* $\mathbf{R}'$*, of which at most one wire could be under the control of the adversary. Then there does not exist any APSMT protocol between* $\mathbf{S}'$ *and* $\mathbf{R}'$*.*

PROOF: The proof is by contradiction. Let $\mathbf{S}'$ and $\mathbf{R}'$ be connected by three bi-directional wires w_1, w_2, w_3, of which at most one wire can be under the control of adversary $\mathcal{A}_1^{static}$. Moreover, let there exists an APSMT protocol Π between $\mathbf{S}'$ and $\mathbf{R}'$ tolerating $\mathcal{A}_1^{static}$. Let E be an execution of Π. Then we define the following variables:

1. $time(E, \mathbf{R}', w_i)$: denotes the arrival time of the different messages (with respect to local clock) received by $\mathbf{R}'$ along wire $w_i, i \in \{1, 2, 3\}$ in E.
2. $time(E, \mathbf{S}', w_i)$: denotes the arrival time of the different messages (with respect to local clock) received by $\mathbf{S}'$ along wire $w_i, i \in \{1, 2, 3\}$ in E.
3. E^{time}: denotes the total time taken (with respect to $\mathbf{R}'$) by execution E; i.e., the time at which $\mathbf{R}'$ terminates by outputting the message in E.

From the termination property of APSMT, each execution of Π will eventually terminate. Moreover, in any execution of Π, the distribution of data sent along a single wire will be same, irrespective of the secret message (which is sent by Π). Otherwise, the adversary can passively listen the wire and will get information about the secret message, thus violating the perfect secrecy property of Π. Now consider the following execution sequences of protocol Π:

1. E_1: The random coin tosses of $\mathbf{S}'$ and $\mathbf{R}'$ are r_1 and r_2 respectively. $\mathbf{S}'$ wants to send the secret m. The adversary strategy is to control wire w_3 and not allowing any data to pass over w_3 throughout E_1. Let α and β denote the messages that are exchanged between $\mathbf{S}'$ and $\mathbf{R}'$, along w_1 and w_2 respectively. The protocol terminates at time E_1^{time}, outputting m.
2. E_2: The random coin tosses of $\mathbf{S}'$ and $\mathbf{R}'$ are r_1 and r_2 respectively. $\mathbf{S}'$ wants to send the secret message m. The adversary strategy is to passively control w_2 and delay any information along w_3 for time $E_1^{time} + E_3^{time} + 1$ (E_3 is defined below). In addition, the adversary schedules the messages along w_1 and w_2 in such a way that $time(E_2, \mathbf{S}', w_i) = time(E_1, \mathbf{S}', w_i)$, for $i \in \{1, 2\}$ and $time(E_2, \mathbf{R}', w_i) = time(E_1, \mathbf{R}', w_i)$, for $i \in \{1, 2\}$. Thus the view of $\mathbf{S}'$ and $\mathbf{R}'$ in E_1 and E_2 are exactly same and hence the secret m is reconstructed. Also $E_1^{time} = E_2^{time}$ and α and β are exchanged between $\mathbf{S}'$ and $\mathbf{R}'$, along w_1 and w_2 respectively.

Let $m^* (\neq m)$ be another secret message. Then from the perfect secrecy property of Π, there exists $r_3 (\neq r_1)$ and $r_4 (\neq r_2)$, such that the following holds: $\mathbf{S}'$ wants to send m^*, the random coin tosses of $\mathbf{S}'$ and $\mathbf{R}'$ are r_3 and r_4 respectively and the information exchanged between $\mathbf{S}'$ and $\mathbf{R}'$ along wire w_2 is β. Note that such an r_3, r_4 exists, otherwise it implies that data sent along wire w_2 is dependent on secret message, thus violating perfect secrecy property of Π. Now consider the following executions of Π:

3. E_3: The random coin tosses of $\mathbf{S}'$ and $\mathbf{R}'$ are r_3 and r_4 respectively. $\mathbf{S}'$ wants to send the secret message m^*. The adversary strategy is to control

wire w_3 and not allowing any data to pass over w_3 throughout E_3. Let α^* and $\beta^*(=\beta)$ denote the messages that are exchanged between $\mathbf{S}'$ and $\mathbf{R}'$, along w_1 and w_2 respectively and the protocol terminates at time E_3^{time}, outputting m^*.

4. E_4: The random coin tosses of $\mathbf{S}'$ and $\mathbf{R}'$ are r_3 and r_4 respectively. $\mathbf{S}'$ wants to send the secret message m^*. The adversary strategy is to passively control w_2 and delay any information along w_3 for time $E_1^{time} + E_3^{time} + 1$. In addition, the adversary schedules the messages along w_1 and w_2 in such a way that $time(E_4, \mathbf{S}', w_i) = time(E_3, \mathbf{S}', w_i)$, for $i \in \{1, 2\}$ and $time(E_4, \mathbf{R}', w_i) = time(E_3, \mathbf{R}', w_i)$, for $i \in \{1, 2\}$. Thus the view of $\mathbf{S}'$ and $\mathbf{R}'$ in E_3 and E_4 are same and hence the secret m^* is reconstructed. Also $E_3^{time} = E_4^{time}$ and α^* and $\beta^*(=\beta)$ are exchanged between $\mathbf{S}'$ and $\mathbf{R}'$, along w_1 and w_2 respectively.

5. E_5: The random coin tosses of $\mathbf{S}'$ and $\mathbf{R}'$ are r_1 and r_4 respectively. $\mathbf{S}'$ wants to send the secret message m. Let $\alpha', \beta'(=\beta)$ denote the messages that should have been exchanged between $\mathbf{S}'$ and $\mathbf{R}'$ along w_1 and w_2 in ideal situation, when w_1 and w_2 are honest (not under the control of adversary).

   Now the adversary strategy in E_5 is as follows: adversary delay any information along w_3 for time $E_1^{time} + E_3^{time} + 1$. In addition, the adversary controls w_1 in Byzantine fashion, such that instead of receiving messages from α', $\mathbf{R}'$ gets messages from α^*, while $\mathbf{S}'$ receives messages from α. Moreover, adversary schedules the messages along w_1 and w_2 in such a way that $time(E_5, \mathbf{S}', w_i) = time(E_2, \mathbf{S}', w_i)$, for $i \in \{1, 2\}$ and $time(E_5, \mathbf{R}', w_i) = time(E_4, \mathbf{R}', w_i)$, for $i \in \{1, 2\}$. Thus the view of $\mathbf{S}'$ is $\alpha\ \beta' = \alpha\ \beta$, while view of $\mathbf{R}'$ is $\alpha^*\ \beta' = \alpha^*\ \beta$.

Thus the view of $\mathbf{S}'$ in E_2 and E_5 are same, so $\mathbf{S}'$ will assume that m has been communicated securely. However, the view of $\mathbf{R}'$ in E_5 is same as in E_4 and hence $\mathbf{R}'$ will output m^*. But this violates the perfect reliability property of Π, which is a contradiction. Hence Π does not exist. □

Theorem 7. *Let* $\mathbf{S}$ *and* $\mathbf{R}$ *be connected by* n *bi-directional wires, of which at most* t *can be under the control of* $\mathcal{A}_t^{static}$*. Then there exits an APSMT protocol tolerating* $\mathcal{A}_t^{static}$ *only if* $n > 3t$*.*

Proof: The proof is by contradiction. Assume that there exist an APSMT protocol Π^{APSMT} between $\mathbf{S}$ and $\mathbf{R}$ tolerating $\mathcal{A}_t^{static}$, where $\mathbf{S}$ and $\mathbf{R}$ are connected by $n = 3t$ bi-directional wires. Now by using standard player partitioning strategy, we show how to transform protocol Π^{APSMT} into another APSMT protocol Π between a sender $\mathbf{S}'$ and a receiver $\mathbf{R}'$, who are connected by three bi-directional wires, of which at most one could be corrupted by the adversary. Let the wires between $\mathbf{S}$ and $\mathbf{R}$ be numbered $1, 2, \ldots, 3t$. Similarly, let the wires between $\mathbf{S}'$ and $\mathbf{R}'$ be numbered as $1, 2, 3$. Now we define a mapping

$M : \{1 \dots n\} \longrightarrow \{1, 2, 3\}$ as follows:

$$\begin{aligned} M(x) &= 1 : \forall x \in \{1 \dots t\} \\ &= 2 : \forall x \in \{t+1 \dots 2t\} \\ &= 3 : \forall x \in \{2t+1 \dots 3t\} \end{aligned}$$

We denote $M^{-1}(1) = \{1, 2, \dots, t\}$, $M^{-1}(2) = \{t+1, t+2, \dots, 2t\}$ and $M^{-1}(3) = \{2t+1, 2t+2, \dots, 3t\}$. Now Π is obtained from Π^{APSMT} in the following way: if in protocol Π^{APSMT}, $k \in \mathbb{F}$ is sent from $\mathbf{S}$ to $\mathbf{R}$ on wire $w \in \{1, 2, \dots, 3t\}$, then in protocol Π, k is sent from $\mathbf{S}'$ to $\mathbf{R}'$ on wire $M(w)$. We define the transmission from $\mathbf{R}'$ to $\mathbf{S}'$ in a similar fashion. Similarly, if the adversary controls wire $w \in \{1, 2, 3\}$ in protocol Π, then he controls the set $M^{-1}(w)$ in protocol Π^{APSMT}. It can be easily verified that the view of $\mathbf{S}'$ and $\mathbf{R}'$ in Π is same as the view of $\mathbf{S}$ and $\mathbf{R}$ respectively, in protocol Π^{APSMT}. So Π is an APSMT protocol between $\mathbf{S}'$ and $\mathbf{R}'$, who are connected by three bi-directional wires, of which at most one can be corrupted. But from Theorem 6, Π does not exist. Hence Π^{APSMT} also does not exist. □

6. ASSMT When All Wires are Unidirectional from S to R

We now give the characterization for ASSMT protocols tolerating $\mathcal{A}_t^{static}$, when all the wires are directed from $\mathbf{S}$ to $\mathbf{R}$.

Theorem 8. *Let there exists n wires directed from $\mathbf{S}$ to $\mathbf{R}$, of which at most t could be under the control of $\mathcal{A}_t^{static}$. Then there exists an ASSMT protocol tolerating $\mathcal{A}_t^{static}$, only if $n > 2t$.*

Proof: From [14, 23], we know that $n > 2t$ wires are necessary for the existence of any synchronous SSMT protocol tolerating an all powerful t-active Byzantine adversary, when all the wires are unidirectional from $\mathbf{S}$ to $\mathbf{R}$. Hence it is obviously necessary for the existence of ASSMT protocol tolerating $\mathcal{A}_t^{static}$, if all the wires are unidirectional from $\mathbf{S}$ to $\mathbf{R}$. □

We now show that $n = 2t + 1$ unidirectional wires from $\mathbf{S}$ to $\mathbf{R}$ are sufficient to design an ASSMT protocol tolerating $\mathcal{A}_t^{static}$. Let $\mathbf{S}$ and $\mathbf{R}$ be connected by $n = 2t + 1$ unidirectional wires, directed from $\mathbf{S}$ to $\mathbf{R}$. Let the wires be denoted by $w_1, \dots, w_n$. Moreover, let $\mathbb{F} = GF(2^\kappa)$, where κ is the error parameter. Furthermore, without loss of generality, let $n = \text{poly}(\kappa)$. We now present an ASSMT protocol called $\Pi_{ASSMT}^{Unidirectional}$, which securely sends a message $m \in \mathbb{F}$. Before describing the protocol, we present a well know tool, which is used in existing PSMT protocols.

Definition 3 (Shamir Secret Sharing [39]). *Let $s \in \mathbb{F}$ be a secret. Then Shamir secret sharing allows to generate n shares of s, such that s can be reconstructed from any $t+1$ shares while any set of t or less shares will not give any*

information about s. This can be done as follows: let $p(x)$ be a random polynomial of degree t over $\mathbb{F}$, such that $p(0) = s$. Let $s_i = p(i)$, for $i = 1, \ldots, n$. Then $s_1, \ldots, s_n$ are called the n shares of s.

It is easy to see that any $t+1$ distinct shares will uniquely reconstruct back the secret s. This is because the $t+1$ shares are nothing but $t+1$ distinct points on $p(x)$, which is of degree t. So using Lagrange interpolation, $p(x)$ and hence $p(0) = s$ can be uniquely reconstructed from $t+1$ shares. On the other hand, s cannot be uniquely reconstructed from t or less shares.

We now present another well know tool, used in existing SSMT protocols.

Definition 4 (Information Theoretically Secure Authentication [36]). *Let $a, b, M \in \mathbb{F}$, where M is the message and a, b are the authentication keys. We define $auth(M; a, b) = aM + b$. It is easy to see that given $auth(M; a, b)$ and M, no information about a and b can be inferred if a and b are unknown. Similarly, given $auth(M; a, b)$ and M, it is not possible to correctly generate $auth(M'; a, b)$, for $M' \neq M$, without knowing a and b, except with probability $\frac{1}{|\mathbb{F}|} \approx 2^{-\Omega(\kappa)}$.*

Protocol $\Pi_{ASSMT}^{Unidirectional}$ is now formally given in Fig. 2.
We now prove the properties of protocol $\Pi_{ASSMT}^{Unidirectional}$.

Claim 1. *In protocol $\Pi_{ASSMT}^{Unidirectional}$, if $\mathbf{R}$ concludes that $p'(i)$ is a valid share, then except with probability $2^{-\Omega(\kappa)}$, $p'(i) = p(i)$.*

Proof: The claim trivially holds without any error if w_i is honest because an honest wire will correctly deliver $p'(i) = p(i)$. So we consider the case when w_i is corrupted. So let w_i be a corrupted wire, who delivers $p'(i) \neq p(i)$. In order that $p'(i)$ is considered as a valid share, it must hold that $Support_i \geq t+1$. This further implies that there exists at least one honest wire, say w_j, such that $w_j \in Support_i$ because there can be at most t corrupted parties. Since $w_j \in Support_i$, it implies that $\gamma'_{ij} = auth(p'(i); a'_{ij}, b'_{ij})$. Now notice that w_j is an honest wire and so $a'_{ij} = a_{ij}$ and $b'_{ij} = b_{ij}$. However $\mathcal{A}_t^{static}$ will have no information about a'_{ij} and b'_{ij}, as they are sent over w_j. So from the properties of $auth$, except with probability $2^{-\Omega(\kappa)}$, $\gamma'_{ij} \neq auth(p'(i); a'_{ij}, b'_{ij})$, which is a contradiction. So except with probability $2^{-\Omega(\kappa)}$, $p'(i) = p(i)$. □

Claim 2. *In protocol $\Pi_{ASSMT}^{Unidirectional}$, $\mathbf{R}$ will eventually get $t+1$ valid shares.*

Proof: In $\Pi_{ASSMT}^{Unidirectional}$, the worst case occurs when at most t corrupted wires do not deliver any information at all. However, still there exists $t+1$ honest wires, who will eventually deliver correct shares to $\mathbf{R}$. These correct shares will eventually reach $\mathbf{R}$ and hence will be considered as valid shares by $\mathbf{R}$. □

Claim 3. *In protocol $\Pi_{ASSMT}^{Unidirectional}$, if $\mathbf{R}$ outputs m', then except with probability $2^{-\Omega(\kappa)}$, $m' = m$.*

Figure 2: Protocol $\Pi_{ASSMT}^{Unidirectional}$ with $n = 2t + 1$ unidirectional wires from **S** to **R**.

Computation and Communication by S:

1. **S** selects a random polynomial $p(x)$ of degree t over $\mathbb{F}$, such that $p(0) = m$, where m is the secret message, which **S** wants to send to **R**.
2. For $i = 1, \ldots, n$, **S** computes $p(i)$.
3. For $i = 1, \ldots, n$, corresponding to $p(i)$, **S** randomly selects n authentication keys $(a_{ij}, b_{ij}) \in \mathbb{F}^2$, for $j = 1, \ldots, n$.
4. For $i = 1, \ldots, n$, **S** computes $\gamma_{ij} = auth(p(i); a_{ij}, b_{ij})$, for $j = 1, \ldots, n$.
5. For $i = 1, \ldots, n$, **S** sends the following to **R** over wire w_i:
   (a) The value $p(i)$;
   (b) γ_{ij}, for $j = 1, \ldots, n$;
   (c) The authentication keys (a_{ji}, b_{ji}), for $j = 1, \ldots, n$.

Message Recovery by R:

For $r = 0, \ldots, t$, **R** does the following in iteration r:

1. Let $\mathcal{W}$ denote the set of wires w_i over which **R** has received a complete set of values; i.e.,
   (a) The value $p'(i)$;
   (b) γ'_{ij}, for $j = 1, \ldots, n$;
   (c) The authentication keys (a'_{ji}, b'_{ji}), for $j = 1, \ldots, n$.

   Let W_r denote the contents of $\mathcal{W}$, when $\mathcal{W}$ contains exactly $t + 1 + r$ wires.
2. Wait until $|\mathcal{W}| \geq t + 1 + r$. Now corresponding to every $w_i \in W_r$, **R** computes

$$Support_i = \{w_j \in W_r : \gamma'_{ij} = auth(p'(i); a'_{ij}, b'_{ij})\}$$

3. If $Support_i \geq t + 1$, then **R** concludes that $p'(i)$ is a valid share.
4. If **R** finds $t + 1$ valid shares, then using them **R** interpolates the t degree polynomial $p'(x)$, output $m' = p'(0)$ and terminates the protocol. Otherwise **R** proceeds to next iteration.

PROOF: If $\mathbf{R}$ outputs m', then it implies that $\mathbf{R}$ must have received $t+1$ valid shares, using which $\mathbf{R}$ has interpolated t degree polynomial $p'(x)$, such that $p'(0) = m'$. In the worst case, out of these $t+1$ valid shares, at most t shares could have been received over the wires which are under the control of $\mathcal{A}_t^{static}$. However, from Claim 1, the probability that none of those t shares are the original shares of m is at most $t2^{-\Omega(\kappa)} \approx 2^{-\Omega(k)}$. So except with probability $2^{-\Omega(k)}$, all the $t+1$ valid shares are indeed the original shares of m. So $m' = m$, except with probability $2^{-\Omega(\kappa)}$. □

Claim 4. *In protocol $\Pi_{ASSMT}^{Unidirectional}$, $\mathcal{A}_t^{static}$ will get no information about m.*

PROOF: Without loss of generality, let $w_1, \ldots, w_t$ be under the control of $\mathcal{A}_t^{static}$. So $\mathcal{A}_t^{static}$ will know $p(1), \ldots, p(t)$. $\mathcal{A}_t^{static}$ will also know the authentication keys (a_{ji}, b_{ji}), for $j = 1, \ldots, n$ and $i = 1, \ldots, t$. But since the authentication keys used to authenticate each share are completely random and independent of each other, they do not provide any extra information to $\mathcal{A}_t^{static}$ about $p(t+1), \ldots, p(n)$. Thus adversary will lack by one point to uniquely interpolate $p(x)$ and so from the properties of Shamir secret sharing [39], $p(0) = m$ will be information theoretically secure. □

Theorem 9. *If there are $n = 2t+1$ unidirectional wires from $\mathbf{S}$ to $\mathbf{R}$, then there exists an efficient ASSMT protocol tolerating $\mathcal{A}_t^{static}$.*

PROOF: Follows from protocol $\Pi_{ASSMT}^{Unidirectional}$ and Claim 1, Claim 2, Claim 3 and Claim 4. □

Theorem 10. *Let $\mathbf{S}$ and $\mathbf{R}$ be connected by n unidirectional wires, directed from $\mathbf{S}$ to $\mathbf{R}$. Then ASSMT tolerating $\mathcal{A}_t^{static}$ is possible iff $n > 2t$.*

PROOF: Follows from Theorem 8 and Theorem 9. □

7. ASSMT When All Wires are Bidirectional Between S to R

The characterization for ASSMT tolerating $\mathcal{A}_t^{static}$, when all the n wires between $\mathbf{S}$ and $\mathbf{R}$ are bi-directional is given by following theorem:

Theorem 11. *Let $\mathbf{S}$ and $\mathbf{R}$ be connected by n bi-directional wires, of which at most t could be under the control of $\mathcal{A}_t^{static}$. Then there exists an ASSMT protocol tolerating $\mathcal{A}_t^{static}$ iff $n > 2t$.*

PROOF: Any bi-directional wire between $\mathbf{S}$ and $\mathbf{R}$ can be treated as an unidirectional wire from $\mathbf{S}$ to $\mathbf{R}$. Now we know that there exists an ASSMT protocol $\Pi_{ASSMT}^{Unidirectional}$ tolerating $\mathcal{A}_t^{static}$ if there exists $n = 2t+1$ unidirectional wires from $\mathbf{S}$ to $\mathbf{R}$. Hence the same protocol can also be executed if there exists $n = 2t+1$ bi-directional wires between $\mathbf{S}$ to $\mathbf{R}$. This proves the sufficiency part.

From [18], we know that $n > 2t$ wires are necessary for the existence of any synchronous SSMT protocol tolerating an all powerful t-active Byzantine adversary, when all the wires are bi-directional. Hence it is obviously necessary for the existence of ASSMT protocol tolerating $\mathcal{A}_t^{static}$, if all the wires between $\mathbf{S}$ and $\mathbf{R}$ bi-directional. □

8. Conclusion and Open Problems

In this paper, we have studied PSMT and SSMT in asynchronous networks. We showed that the existing PSMT protocol of [38] does not provide perfect secrecy. We then give the exact characterization of PSMT in asynchronous networks. We also give the necessary and sufficient condition for SSMT in asynchronous networks. Our characterization reveals that asynchrony of the network significantly affects the connectivity requirement for PSMT, where as asynchrony does not play any role in determining the connectivity requirement for SSMT.

In this paper we have considered two network settings: one, when all the wires between **S** and **R** are directed from **S** to **R** and second, when all the wires between **S** and **R** are bidirectional. It would be interesting to consider a more general case, when certain wires are directed from **S** to **R** and certain wires are directed from **R** to **S**. Finding the characterization of PSMT and SSMT tolerating mixed adversary in asynchronous network is yet another interesting problem.

References

[1] I. Abraham, D. Dolev, and J. Y. Halpern. An almost-surely terminating polynomial protocol for asynchronous Byzantine Agreement with optimal resilience. 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*, pages 405–414. ACM Press, 2008.

[2] 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.

[3] D. Beaver. Efficient Multiparty Protocols Using Circuit Randomization. In J. Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 420–432, 1992.

[4] Z. Beerliová-Trubíniová and M. Hirt. Efficient Multi-party Computation with Dispute Control. In S. Halevi and T. Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 305–328. Springer, 2006.

[5] Z. Beerliová-Trubíniová and M. Hirt. Simple and efficient perfectly-secure asynchronous MPC. In K. Kurosawa, editor, *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings*, volume 4833 of *Lecture Notes in Computer Science*, pages 376–392. Springer Verlag, 2007.

[6] Z. Beerliová-Trubíniová and M. Hirt. Perfectly-Secure MPC with Linear Communication Complexity. In R. Canetti, editor, *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008*, volume 4948 of *Lecture Notes in Computer Science*, pages 213–230. Springer, 2008.

[7] M. Ben-Or, R. Canetti, and O. Goldreich. Asynchronous secure computation. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, 1993*, pages 52–61. ACM Press, 1993.

[8] 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.

[9] M. BenOr, B. Kelmer, and T. Rabin. Asynchronous secure computations with optimal resilience. In *Proceedings of the Thirteenth Annual ACM Symposium on Principles of Distributed Computing, Los Angeles, California, USA, August 14-17*, pages 183–192. ACM Press, 1994.

[10] R. Canetti. *Studies in Secure Multiparty Computation and Applications*. PhD thesis, Weizmann Institute, Israel, 1995.

[11] R. Canetti and T. Rabin. Fast Asynchronous Byzantine Agreement with Optimal Resilience. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, 1993*, pages 42–51. ACM, 1993.

[12] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults (Extended Abstract). In *26th Annual Symposium on Foundations of Computer Science, 21-23 October 1985, Portland, Oregon, USA*, pages 383–395. IEEE, 1985.

[13] 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.

[14] 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.

[15] D. Dolev. The Byzantine Generals Strike Again. *Journal of Algorithms*, 3:14–30, 1982.

[16] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly Secure Message Transmission. *JACM*, 40(1):17–47, 1993.

[17] M. Fitzi, M. K. Franklin, J. A. Garay, and S. Harsha 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.

[18] M. Franklin and R. Wright. Secure Communication in Minimal Connectivity Models. *Journal of Cryptology*, 13(1):9–30, 2000.

[19] R. Gennaro, Y. Ishai, E. Kushilevitz, and T. Rabin. The Round Complexity of Verifiable Secret Sharing and Secure Multicast. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 580–589. ACM, 2001.

[20] O. Goldreich, S. Micali, and A. Wigderson. How to Play Any Mental Game. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, USA*, pages 218–229. ACM, 1987.

[21] M. Hirt, U. Maurer, and B. Przydatek. Efficient secure multiparty computation. In T. Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*, pages 143–161. Springer Verlag, 2000.

[22] 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.

[23] K. Kurosawa and K. Suzuki. Almost secure (1-round, n-channel) message transmission scheme. *IEICE Transactions*, 92-A(1):105–112, 2009.

[24] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[25] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. North-Holland Publishing Company, 1978.

[26] K. Menger. Zur Allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10:96–115, 1927.

[27] A. Patra, A. Choudhary, T. Rabin, and C. Pandu Rangan. The round complexity of verifiable secret sharing revisited. In S. Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 487–504. Springer Verlag, 2009.

[28] A. Patra, A. Choudhary, and C. Pandu Rangan. Efficient Statistical Asynchronous Verifiable Secret Sharing with Optimal Resilience. In K. Kurosawa, editor, *Information Theoretic Security, Fourth International Conference, ICITS 2009, Shizuoka, Japan, December 3-6, 2009, Proceedings*, volume 5973 of *Lecture Notes in Computer Science*, pages 74–92. Springer Verlag, 2009.

[29] A. Patra, A. Choudhary, and C. Pandu Rangan. Communication Efficient Perfectly Secure VSS and MPC in Asynchronous Networks with Optimal Resilience. In D.J. Bernstein and T. Lange, editors, *Advances in Cryptology - AFRICACRYPT'10, Third International Conference in Cryptology in Africa, Stellenbosch, South Africa, May 3-6, 2009, Proceedings*, volume 6055 of *Lecture Notes in Computer Science*, pages 184–202. Springer Verlag, 2010.

[30] 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.

[31] A. Patra, A. Choudhury, and C. Pandu Rangan. Efficient asynchronous Byzantine agreement with optimal resilience. Cryptology ePrint Archive, Report 2008/424, 2008. A preliminary version of this paper appeared in PODC 2009.

[32] A. Patra, A. Choudhury, K. Srinathan, and C. Pandu Rangan. Unconditionally reliable and secure message transmission in undirected synchronous networks: Possibility, feasibility and optimality. *International Journal of Applied Cryptography*, 2(2):159–197, 2010.

[33] A. Patra and C. Pandu Rangan. Communication optimal multi-valued asynchronous broadcast and asynchronous Byzantine agreement. Cryptology ePrint Archive, Report 2009/433, 2009. To appear in ICITS 2011.

[34] M. Pease, R. E. Shostak, and L. Lamport. Reaching Agreement in the Presence of Faults. *JACM*, 27(2):228–234, 1980.

[35] B. Prabhu, K. Srinathan, and C. Pandu Rangan. Trading players for efficiency in unconditional multiparty computation. In S. Cimato, C. Galdi, and G. Persiano, editors, *Security in Communication Networks, Third International Conference, SCN 2002, Amalfi, Italy, September 11-13, 2002. Revised Papers*, volume 2576 of *Lecture Notes in Computer Science*, pages 342–353. Springer Verlag, 2002.

[36] T. Rabin and M. Ben-Or. Verifiable Secret Sharing and Multiparty Protocols with Honest Majority (Extended Abstract). In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 73–85. ACM, 1989.

[37] 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.

[38] H. Sayeed and H. Abu-Amara. Efficient Perfectly Secure Message Transmission in Synchronous Networks. *Information and Computation*, 126(1):53–61, 1996.

[39] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[40] 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.

[41] K. Srinathan, M. V. N. Ashwin Kumar, and C. Pandu Rangan. Asynchronous Secure Communication Tolerating Mixed Adversaries. In Yuliang Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, volume 2501 of *Lecture Notes in Computer Science*, pages 224–242. Springer, 2002.

[42] 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.

[43] K. Srinathan and C. Pandu Rangan. Efficient asynchronous secure multiparty distributed computation. In B. K. Roy and E. Okamoto, editors,

Progress in Cryptology - INDOCRYPT 2000, First International Conference in Cryptology in India, Calcutta, India, December 10-13, 2000, Proceedings, volume 1977 of *Lecture Notes in Computer Science*, pages 117–129. Springer Verlag, 2000.

[44] A. C. Yao. Protocols for Secure Computations. In *Proceedings of 23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, 3-5 November 1982*, pages 160–164. IEEE Computer Scciety, 1982.