Secure Message Transmission in the General Adversary Model

Qiushi Yang

A thesis submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy
of
University College London

September 18, 2011

Department of Computer Science
University College London

Declaration:

I hereby declare that this submission is the result of my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher education, except where explicitly indicated in the text.

Qiushi Yang	**Date**

Abstract

The problem of *secure message transmission* (SMT), due to its importance in both practice and theory, has been studied extensively. Given a communication network in which a sender S and a receiver R are indirectly connected by unreliable and distrusted channels, the aim of SMT is to enable messages to be transmitted from S to R with a reasonably high level of *privacy* and *reliability.* SMT must be achieved in the presence of a *Byzantine adversary* who has unlimited computational power and can corrupt the transmission. In the *general adversary model*, the adversary is characterized by an *adversary structure.* We study two different measures of security: *perfect* (PSMT) and *almost perfect* (APSMT). Moreover, *reliable* (but not private) *message transmission* (RMT) are considered as a specific part of SMT. In this thesis, we study RMT, APSMT and PSMT in two different network settings: *point-to-point* and *multicast.*

To prepare the study of SMT in these two network settings, we present some ideas and observations on *secret sharing schemes* (SSSs), *generalized linear codes* and *critical paths.* First, we prove that the error-correcting capability of an almost perfect SSS is the same as a perfect SSS. Next, we regard general access structures as linear codes, and introduce some new properties that allow us to construct *pseudo-basis* for efficient PSMT protocol design. In addition, we define adversary structures over "critical paths", and observe their properties. Having these new developments, the contributions on SMT in the aforementioned two network settings can be presented as follows.

The results on SMT in *point-to-point* networks are obtained in three aspects. First, we show a *Guessing Attack* on some existing PSMT protocols. This attack is critically important to the design of PSMT protocols in asymmetric networks. Second, we determine necessary and sufficient conditions for different levels of RMT and APSMT. In particular, by applying the result on almost perfect SSS, we show that relaxing the requirement of privacy does not weaken the minimal network connectivity. Our final contribution in the point-to-point model is to give the first ever efficient, constant round PSMT protocols in the general adversary model. These protocols are designed using linear codes and critical paths, and they significantly improve some previous results in terms of *communication complexity* and *round complexity.*

Regarding SMT in *multicast* networks, we solve a problem that has been open for over a decade. That is, we show the necessary and sufficient conditions for all levels of SMT in different adversary models. First, we give an *Extended Characterization* of the network graphs based on our observation on the *eavesdropping* and *separating* activities of the adversary. Next, we determine the necessary and sufficient conditions for SMT in the general adversary model with the new Extended Characterization. Finally, we apply the results to the threshold adversary model to completely solve the problem of SMT in general multicast network graphs.

Acknowledgements

I would like to express my most profound gratitude to my supervisor Prof. Yvo Desmedt for giving me the opportunity to start my career as a researcher under his supervision. His invaluable advices and suggestions have always been inspiring and helpful. His support, encouragement and guidance carried me through the hardest times.

I would also like to offer my sincere gratitude to my second supervisor Dr. Jens Groth and my report assessor Dr. Nicolas Courtois. The discussions with them helped me gain a clear insight on the area of cryptography and also helped me figure out the directions of my research.

I want to thank two of my nicest colleagues Stelios Erotokritou and Shah Mahmood, especially Stelios, who is extremely kind and selfless, and has been a great help to my research.

I would like to acknowledge the researchers with whom very useful discussions have been made through email and when attending conferences, namely Kaoru Kurosawa, C. Pandu Rangan, Yongge Wang, Arpita Patra and Ashish Choudhury.

I am forever grateful to my parents for their love and support at all times. They have the patience to listen and the wisdom to give the best advice. I hope I can see them more often.

Last but not least, I would like to thank all my friends in the UK and in China. If not for them, I would have graduated long time ago. Just kidding.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Problems of secure communication and computation in different network models have been studied extensively over the last decades. In a network, it is obvious that secure communication between two parties is guaranteed if they are connected by a private and authenticated channel. However, in most cases, many parties are only indirectly connected. That is, most parties need to use intermediate parties in order to communicate with each other. This kind of network can be seen as an incomplete graph, in which the nodes are the parties and the edges are authenticated communication channels. Since some of the intermediate parties may be corrupted by a malicious force, the channels between two indirectly connected parties are distrusted. In this thesis, we study *secure message transmission* (SMT) in such networks. More specifically, we study SMT in the so-called *general adversary model*, in which the corrupted parties are characterized by an *adversary structure*.

In this chapter, we introduce the problems that we are going to deal with, and briefly describe our contributions. In the following Section 1.1, we show the basics of the problem of SMT and reveal the motivation of our research. Since we are particularly interested in the general adversary model, we define and discuss this model in Section 1.2. In Section 1.3, we briefly undergo a survey on the previous studies, and discuss the problems this work will solve. The contributions of this thesis are presented in Section 1.4, which is followed by an overview of the organization of this thesis in Section 1.5.

1.1 Secure Message Transmission

The problem of secure message transmission (SMT) uses the following setting: a party, namely a *sender* S, wants to send to another party, namely a *receiver* R, one or more messages. The distrust of the network is modelled by an entity called *adversary*, who is active and adaptive, and can control some parties in the network with unbounded computational power. The goal of SMT is to allow the messages to be transmitted with *privacy* and *reliability*; i.e., to guarantee some level of secrecy and integrity of the messages when the receiver R receives them, despite the presence of an adversary. This setting uses the *information-theoretic* security approach (Shannon security), which means that the cryptographic methods based on computational hardness cannot be applied because of the unbounded computational power that the adversary possesses. In addition, it is assumed that no secret key is shared between the sender S and the receiver R before the transmission.

To make SMT possible, the corrupting ability of the adversary must be limited. There are two approaches to characterize an adversary: *the threshold model* and *the general adversary model.* The threshold model assumes that any parties can be taken over by the adversary, but the number of the corrupted parties is bounded by a threshold t; i.e., the adversary can control up to t parties. On the other hand, in the general adversary model [ISN87, HM00], the adversary is characterized by an *adversary structure*, which is a set of subsets of the parties in the network; i.e., the adversary can control a subset of parties in the adversary structure. It is worth noting that an adversary structure is relatively more general than a threshold in the context, as we shall discuss in the following Section 1.2.

The SMT setting can find many applications in practice and in theory. Concerning information security, SMT solutions can be practically applied on telephone networks, TV and radio networks, wireless communication, the Internet, etc. For example, Desmedt [Des05] (see also [Des06]) motivated the use of SMT on the Internet by showing that potential attacks against routers can compromise the integrity of the Internet. Indeed, current Internet protocols, including IPsec, do not have the resilience to deal with the corrupted routers, which may completely disrupt the communication in a malicious manner. Therefore, in order to achieve reliable communication on the Internet against attacks on routers, the SMT setting can be applied in such a manner that the parties are used to model the routers. Thus resilient Internet connectivity can be achieved.

Theoretically, not only is SMT important in its own right, but it is also an essential primitive for achieving secure distributed computation. For example, most studies of secure multiparty computation (MPC) assume that there is a secure communication channel between each pair of players (parties) (see, e.g., [BOGW88, CCD88, GRR98, FHM99, CDM00]). It is clear that for MPC in network models, SMT is essential for achieving secure communication between the players. Another motivation of studying SMT is to achieve information-theoretic security. The security of all existing cryptographic techniques (e.g., AES), including public key cryptosystems (e.g., RSA), is based on unproven hardness assumptions. These assumptions can be weakened by factors such as the increase in computing speed and the advent of new computing paradigms (e.g., Quantum computing [Sho97]). Thus achieving information-theoretic security is important because it cannot be affected by the factors mentioned above.

For different applications, different levels of security are required. As in [FW98], two

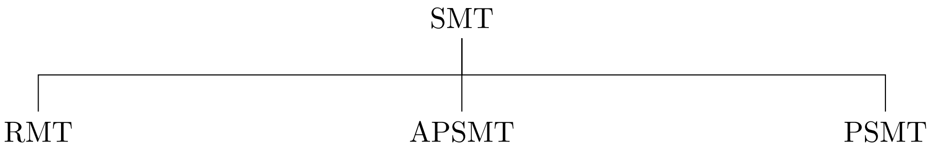


Figure 1.1: Problem set of SMT.

different measures of SMT are considered: *perfect* security (PSMT; i.e., zero probability that the transmission fails to be secure) and *almost perfect* security (APSMT; i.e., an arbitrarily small probability that the transmission fails to be secure). It is important to note that security in this context is measured by both privacy and reliability. Thus PSMT requires both perfect privacy (i.e., the adversary learns absolutely no information about the messages) and perfect reliability (i.e., the receiver R outputs the messages with complete correctness). On the other hand, APSMT allows non-perfection on either privacy or reliability, or both.

In most cases, reliability is important in its own right. For example, when a message is opened to all the parties, no privacy is required, but *reliable message transmission* (RMT) must be enabled from the sender to each receiver. In all SMT scenarios, RMT is essential, though privacy may not always be needed. Thus we study RMT as a specific part of SMT. Therefore, the problem set of SMT can be shown in Figure 1.1.[1]

The advantage of SMT is that it achieves information-theoretic security, which is the strongest notion of security. However, due to the lack of cryptographic methods being used, achieving SMT usually requires relatively strong network connectivity and high communication cost. The communication cost of an SMT protocol is normally calculated in *communication complexity* (CC) [Yao79], which is the number of bits transmitted over the network through the communication. However, *round complexity* (RC) [DDWY93] is constantly taken into consideration as well. Round complexity is measured by the number of rounds taken by an SMT protocol, where a round is a communication from S to R or vice versa. In different network settings and adversary models, for different security and communication requirements, the study of SMT normally aims for two different kinds of contributions:

1. To determine network connectivity; i.e., to find the *necessary and sufficient* conditions that the network must satisfy in order to allow for the existence of an SMT protocol.

2. To minimize communication cost; i.e., to design *efficient* SMT protocols with toward-optimal communication cost in terms of CC and RC.[2]

[1] Note that the difference between RMT and APSMT is that RMT guarantees no privacy whatsoever, while APSMT requires perfect or almost perfect privacy. We extend this problem set in our security model in Section 2.4.

[2] Normally, a protocol is called *efficient* if its CC is polynomial in the number of communication channels in a network. Most studies tend to design efficient protocols that are executed in a constant number of rounds (i.e., constant RC).

In this thesis, we make both kinds of contributions as mentioned above, mainly in the *general adversary model*. The motivation of studying this model is presented in the following section.

1.2 The General Adversary Model

In the general adversary model, the adversary is characterized by an *adversary structure*. The concept of adversary structures is based on the concept of access structures [ISN87], and was first proposed by Hirt and Maurer [HM00] in the context of secure multiparty computation. An adversary structure can be defined as follows:

Definition 1.2.1. (see [HM00]) *Given a party set D, an adversary structure $\mathcal{A}$ on D is a family of subsets $\mathcal{A} \subseteq 2^D$ such that for any $A \in 2^D$, if $A \in \mathcal{A}$ and $A' \subseteq A$, then $A' \in \mathcal{A}$.*

Obviously, an adversary structure $\mathcal{A}$ is by definition monotone, thus we define the *basis* of $\mathcal{A}$ as a set $\widehat{\mathcal{A}} \subseteq \mathcal{A}$ such that if and only if a set $A \in \widehat{\mathcal{A}}$, then $\forall A' \supsetneq A : A' \notin \mathcal{A}$. In general, the number of members in $\mathcal{A}$ or $\widehat{\mathcal{A}}$ is exponential in the number of members in D.

As an example, let $D = \{1, 2, 3\}$ be a party set and

$$\mathcal{A} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}\}$$

be an adversary structure, the adversary can choose to control any *one* set from $\mathcal{A}$; e.g., if it chooses $\{1, 2\}$ to control, then it is unable to control party 3. Note that if $\{2, 3\}$ was in $\mathcal{A}$, then the adversary would be 2-bounded—just as in the threshold model. This is because the adversary could choose to control any 2 parties. Thus it is clear that a threshold t is a special adversary structure, which consists of all subsets that contain at most t parties. Since an adversary structure has a non-threshold meaning in most cases, it is more general than a threshold.

Burmester and Desmedt [BD04] motivated this model by introducing the common platform attacks. The idea is that a hacker who can exploit a weakness in one platform, can with almost the same ease attack many computers on the same platform. One obvious example is the computer viruses, such as the Internet virus and worms that only spread on Windows, or sometimes Unix. To model such an attack, the general adversary model is clearly more appropriate than the threshold model.

On the other hand, due to the generality of the general adversary model, the research in this model can be applied directly in the threshold model. That is, even in the study of the threshold model, the work can be done in such a manner that an adversary structure is studied, and the results can be straightforwardly used to find the respective results in the threshold model. Therefore, the study of the general adversary model is obviously important, and it has received a considerable amount of attention in the research of secure multiparty computation (see, e.g., [HM00, BW98, FHM99, CDM00, DDFN07]).

In the following section, we briefly review the previous work in the field of SMT research regarding both the threshold and general adversary models. We shall provide a more detailed survey on the related previous results later in Section 2.9 after we give all the definitions and models. Therefore, Section 1.3 could be seen as a problem statement, and Section 2.9 is a more detailed summary of the previous results.

1.3 Previous Work (in Brief)

Given a sender S and a receiver R, SMT between S and R has been studied with different kinds of network, adversary and security models. The contributions of the previous work included the determination of minimal network connectivities and the construction of efficient protocols with minimized communication cost.

Most studies of SMT considered the threshold model. The initial study of PSMT was carried out by Dolev et al. in [DDWY93] (see also [Dol82]). Their work assumed a threshold adversary in an *undirected* synchronous network, and showed the fundamental condition for PSMT in such a network. In the study by Desmedt and Wang [DW02] (see also [WD08]), *directed* networks were considered. Their setting allows asymmetric communications between S and R by introducing the concept of feedback channels (i.e., channels from R to S). Their work showed the necessary and sufficient condition for PSMT in this network setting.

The concept of APSMT was introduced by Franklin and Wright in [FW98] (see also [FW00]). They showed that Dolev et al.'s fundamental condition for PSMT in *undirected* networks is also necessary and sufficient for APSMT. However, a later study by Desmedt and Wang [DW02] showed that in *directed* networks with feedback channels, the network connectivity required for APSMT is weaker than that for PSMT. This problem was further studied by Srinathan and Rangan in [SR06], whose results were later extended by Shankar et al. in [SGSR08]. In all these studies of APSMT, the requirement for reliability was weakened. That is, these results considered perfect privacy and almost perfect reliability. However, there is another kind of APSMT, where the requirement for privacy is lower, but perfect reliability must be guaranteed. *This kind of almost perfect private message transmission has not been considered in previous studies.*

The work to minimize the communication cost of PSMT protocols has been done in a sequence of studies. In the initial study by Dolev et al. [DDWY93], a 2-round PSMT protocol was provided with communication complexity (CC) exponential in the number of communication channels (i.e., n), which is inefficient. Later in [SAA96], Sayeed and Abu-Amara improved the result by showing a 2-round PSMT protocol with CC polynomial in n. More recently, Srinathan et al. [SNR04] gave the lower bound on CC for 2-round PSMT protocols: $\Omega(n\ell)$, where ℓ is the number of bits of the transmitted message. Any protocol that achieves such a CC is thus called *optimal*. However, their proposed optimal 2-round PSMT protocol is flawed, as pointed out by Agarwal et al. [ACdH06], who presented their own 2-round PSMT protocol, which is optimal, but at the price of requiring ℓ to be exponential in n. Later studies by Patra et al. [PCSR06] and Fitzi et al. [FFGS07] both gave optimal PSMT protocols that transmit messages of length polynomial in n. However, the result of [PCSR06] requires an additional round, and the result of [FFGS07] requires additional channels. The problem of designing an optimal 2-round PSMT protocol was finally solved by Kurosawa and Suzuki in [KS08, KS09c]. Their results were obtained based on a new idea of *pseudo-basis* and *pseudo-dimension* (see Appendix A.6). All the above studies considered *undirected* networks. Using the similar technique as that in [KS08], Patra et al. proposed in [PCR10] the optimal 3-round PSMT protocols in *directed* networks.

Most studies on SMT used the *point-to-point* communication channels; i.e., there is only one party at each end of a point-to-point channel. In [FY95], Franklin and Yung initialized the study of SMT via *multicast* (partial broadcast) channels. A multicast channel enables messages to be sent from a party to a fixed subset of parties on this channel. Several different partial broadcast network settings have been introduced by Franklin and Yung [FY95], but they only studied perfect privacy against a *passive* adversary (i.e., an eavesdropper without the capability to corrupt the transmission). A very different full information model was considered by Goldreich et al. in [GGL98]. Later Franklin and Wright [FW98] (see also [FW00]) studied SMT on a specific *neighbour network* setting, in which each party must communicate simultaneously to all its neighbours in the underlying network. Some further studies on this multicast model have been carried out by Wang and Desmedt in [WD99] and Desmedt and Wang in [DW02]. We note that the previous results demonstrated in such multicast neighbour networks are under a strong assumption; i.e., all communication channels between S and R are *neighbour-disjoint*.[3] Indeed Franklin and Wright in [FW98] raised the following open problem:

> "... if these n disjoint paths do not have disjoint neighbourhood, then an adversary may be able to foil our protocols with $t < n$ faults by using one fault to eavesdrop on two disjoint lines. An obvious direction of further research is to characterize secure communication fully in this more general setting."

Therefore, *the minimal network connectivity for SMT in a general multicast network remains unknown for any level of security.*

All the above studies considered the threshold model. In addition, SMT in the general adversary model has also been studied. The research was initiated by Kumar et al. [KGSR02], who showed the necessary and sufficient condition for PSMT

[3] Neighbour-disjoint means that any two communication channels between S and R do not have a common neighbour.

in undirected point-to-point networks. Desmedt et al. [DWB05] showed the minimal network connectivity required for 1-round PSMT protocols, and Patra et al. [PSC$^+$07] presented the necessary and sufficient condition for PSMT in directed networks with feedback channels.

Although all conditions for PSMT have been determined, the research of SMT in the general adversary model is still in the beginning phase, with the following problems remain unsolved:

- What are the necessary and sufficient conditions for RMT and APSMT in point-to-point networks?
- How is it possible to reduce the communication cost of the previous PSMT protocols, which are relatively inefficient in terms of CC and RC?
- What are the necessary and sufficient conditions for SMT (including different levels of RMT, APSMT and PSMT) in general multicast networks?

Our research is devoted to answer these questions. In the following section, we highlight the contributions of this thesis.

1.4 Contributions of the Thesis

The main contributions of this thesis are presented in two aspects: (1) SMT in point-to-point networks and (2) SMT in multicast networks. However, many of our results on SMT are based on our new findings on *secret sharing schemes* (SSSs), *linear codes* and *critical paths*. Thus we discuss these basic ideas and observations before we show the main results.

1.4.1 Ideas and Observations

The first contribution of this thesis relies on some basic ideas and observations on secret sharing, error-correcting, linear codes and critical paths, which are the foundations of our study of SMT. These ideas and observations are presented in three aspects, which we describe as follows.

First, we examine the error-correcting capability of an *almost perfect* secret sharing scheme. Secret sharing schemes (SSSs) are an essential tool in the study of SMT. To study almost perfect security, we analyse the properties of a newly defined almost perfect SSS. Using exhaustive search, we show that an almost perfect SSS can detect and correct the same number of errors as a perfect SSS can. This result will later be used to prove the necessary and sufficient conditions for almost perfect privacy.

Next, we construct a generalized linear code, which is derived from a *linear secret sharing scheme* (LSSS) considering an adversary structure. Previous studies showed that a generalized LSSS can be constructed using a *monotone span program* [KW93, CDM00].

In this thesis we convert the LSSS into a linear code with error-correcting capability. Using the linear code, we extend the idea of *pseudo-basis* and *pseudo-dimension* (see [KS08]) in the general adversary model.

Finally, we show our observation on the *critical paths* between S and R. With the help of some examples, we show that if messages are transmitted via the paths between the sender S and the receiver R, then the number of elements transmitted is independent to the size of the network. Moreover, we claim that the communication complexity (CC) of most SMT protocols is determined by the number of critical paths, instead of the size of the network. Thus we define the *critical-path structure*, which is converted from the adversary structure. This observation should provide a clear view on the problem of SMT in the general adversary model.

With these ideas and observations, we are ready to solve the problems of SMT in point-to-point and multicast networks.

1.4.2 SMT in Point-to-Point Networks

We study SMT in point-to-point networks and make three aspects of contributions.

First, we cryptanalyse some existing PSMT protocols in directed point-to-point networks. In a directed network with feedback channels from R to S, the feedbacks are normally used by the receiver R for reliability purposes when faulty messages are received. We design a *Guessing Attack*, which can be performed by the adversary on the feedback channels to breach perfect privacy of some existing protocols in [DW02, PSC$^+$07]. This kind of attack should always be considered when designing SMT protocols in such a network setting.

Next, we determine minimal network connectivities for different levels of RMT and APSMT in the point-to-point model. First, by generalizing the idea of [DW02], we show the necessary and sufficient conditions for achieving almost perfect reliability in different network settings. Next, using the previously mentioned result regarding the error-correcting capability of an almost perfect SSS, we prove that in order to achieve almost perfect privacy, the same connectivity is required as that for perfect privacy. In other words, reducing the requirement for privacy does not weaken the minimal connectivity. This is sometimes different to some cases when the requirement for reliability is relaxed. These results complete the research of determining minimal connectivities in the point-to-point model, and hence answer the first question raised at the end of Section 1.3. A summary of the minimal network connectivities for SMT in point-to-point networks is shown in Table 6.1 in the concluding Chapter 6 of this thesis.

Our final contribution in the point-to-point model is on PSMT protocols. Using the linear code we discussed earlier, we can design efficient PSMT protocols in constant rounds. Our results significantly improve the previous protocols in terms of communication complexity (CC) and round complexity (RC) (see Table 4.1 in Section 4.3). Indeed, our protocols are the first ever efficient, constant-round PSMT protocols proposed in the general adversary model. Furthermore, we are also the first to study PSMT of multiple messages in this context. These results answer the second question raised at the end of Section 1.3.

1.4.3 SMT in Multicast Networks

As we discussed in Section 1.3, the previous results regarding SMT in multicast neighbour networks require node-disjoint and neighbour-disjoint paths. This requirement is not necessary in the general multicast network graph setting. To that extent the necessary and sufficient conditions for SMT in general multicast networks remain unknown. We completely solve this problem, which has been open for over a decade. Our solution is based on two basic ideas: (1) a general graph setting can be applied naturally in the general adversary model; (2) a threshold corresponds to a special adversary structure (see Section 1.2). Thus we study SMT in multicast neighbour networks in the general adversary model, and then apply the results to the threshold model.

First, we realize that the current adversary structure model is not enough to characterize general multicast networks, so we give an *Extended Characterization* of the network graphs. This characterization is based on our observation on the *eavesdropping* and *separating* activities of the adversary on the multicast channels. This observation allows us to gain a clearer insight on the multicast communication.

Next, using the Extended Characterization, we give the necessary and sufficient conditions for different levels of SMT (including RMT, APSMT and PSMT) in the general adversary model. Besides proving that our conditions imply the lower bounds on the network connectivity, we also provide communication protocols to show that the bounds are tight.

Finally, we use the results in the general adversary model to find the necessary and sufficient conditions for SMT in the threshold model. Additionally, by analysing some previous results (e.g., in [WD99, DW02]), we show how our results explain all the examples and prove all the conjectures in previous work.

The necessary and sufficient conditions for SMT in multicast neighbour networks are the final contributions of this thesis. These results answer the third question raised at the end of Section 1.3. A summary of the minimal network connectivities for SMT in multicast networks is shown in Table 6.1 in Section 6.1.

1.5 Organization of the Thesis

The organization of this thesis strictly follows our description of the contributions in the previous section. That is, we first show “Ideas and Observations”, then “SMT in Point-to-Point Networks”, and finally “SMT in Multicast Networks”. We present the “Model and Background” before showing the results, and summarize our results in “Conclusion and Future Work” at the end of this thesis.

First, in Chapter 2, we give the definitions and notations for our network, adversary and security models, discuss the basics of communication cost, secret sharing schemes, error-correcting codes and authentication codes, and finally summarize the related previous results in more detail. These are the background to our research.

In Chapter 3, we present our ideas and observations. We study the error-correcting capability of a threshold almost perfect SSS (see Section 3.1), which will later be used to determine the minimal connectivities for almost private communication. We also convert the generalized linear secret sharing scheme (LSSS) to a linear code with error-correcting capability, and then generalize the idea of pseudo-basis and pseudo-dimension by using the linear code (see Section 3.2). This result will later be used to design efficient PSMT protocols. Finally, we define adversary structures over critical paths, and observe their properties (see Section 3.3).

In Chapter 4, we study SMT in point-to-point networks. We design a Guessing Attack on the feedback channels to breach perfect privacy of some previous protocols (see Section 4.1). Then we determine the minimal connectivity requirements for all levels of RMT and APSMT in both undirected and directed network graph settings (see Section 4.2). Finally, we propose a number of efficient protocols using the above mentioned linear code, and show how these protocols improve the previous results in the general adversary model. Indeed, our protocols make some significant improvements to the previous results in terms of communication complexity (CC) and round complexity (RC) (see Section 4.3, Section 4.4 and Section 4.5).

In Chapter 5, we study SMT in multicast neighbour networks. First we show our observation on the eavesdropping and separating activities on a single multicast channel, and use our observation to further characterize the multicast graph for our communication model (see Section 5.1). We then study different levels of reliability (RMT) and security (SMT) in the general adversary model (see Section 5.2 and Section 5.3), and finally apply these results in the threshold model to completely solve the problem of SMT in general multicast graphs (see Section 5.4).

Finally, in Chapter 6, we summarize the results of this thesis (see Section 6.1) and raise open problems for future work (see Section 6.2).

Chapter 2

Model and Background

In this chapter we describe the network, adversary and security models, and give definitions and notations to be used in the rest of this thesis. We also present some related results and techniques from previous studies in detail, which are used as the background to our study.

2.1 Basics

We denote $\mathbb{F}$ as a finite field which is assumed to be sufficiently large, and use ρ to denote the length of the field elements. We denote $\mathbb{M} \subseteq \mathbb{F}$ as a message space from which the messages are drawn. Thus each message is a field element of length ρ. Let A be a set, we write $a \in_R A$ to indicate that a is chosen from A with respect to the uniform distribution. Let $a \in \mathbb{R}$, we write $\lfloor a \rfloor \in \mathbb{Z}$ to denote the integer part of a, and $\lceil a \rceil \in \mathbb{Z}$ to denote the smallest integer that is greater than or equal to a.

2.2 Network Model

We model two different kinds of networks: the *point-to-point* networks and the *multicast* networks. A point-to-point network consists of a number of point-to-point channels each of which enables a party to send an element to another party through the channel, while a multicast network consists of a number of multicast (or "partial broadcast") channels each of which enables a party to send an element to a fixed subsets of parties through the channel [FY95]. Besides describing these two network settings in Section 2.2.1 and Section 2.2.2 respectively, we also define different kinds of network connectivities in both the threshold and general adversary models in Section 2.2.3.

2.2.1 Point-to-Point Networks

We abstract away the concrete network structure and model a point-to-point network by a graph $G(V, E)$, whose nodes are the parties in the network and edges are private and authenticated point-to-point channels. Let $S, R \in V$, where S is the sender and R is the receiver.

For different types of communication channels, we use different network graphs:

Undirected graphs. In an undirected graph, all edges in E are undirected, so the communication on the edges can go in both directions. Therefore, the undirected paths allow two-way communication between S and R (see, e.g., [DDWY93]).

Directed graphs. In a directed graph, all edges in E have directions, that is, they are either one-way directed or bi-directed. Therefore, the directed paths between S and R can be distinguished as either *forward paths* (i.e., from S to R) or *feedback paths* (i.e., from R to S) or, otherwise, *heterogeneous paths* (see, e.g., [DW02, SR06]). Note that a path can be both forward and feedback if all edges composing it are bi-directed.

2.2.2 Multicast Networks

In the initial work [FY95], a multicast network is modelled by a hypergraph $H(V, E_H)$, where the nodes in V are the parties in the network and E_H is the set of all hyperedges (v, V^*) where $v \in V$ and $V^* \subseteq V \setminus \{v\}$. Thus, a hyperedge (v, V^*) is a multicast channel on which an element multicast by node v will be received—simultaneously and privately—by all nodes in V^*.

As in [FY95], given two nodes $v_1, v_2 \in V$, we say that there is a *directed link* from v_1 to v_2 if there exists a hyperedge (v_1, V_1^*) such that $v_2 \in V_1^*$, and we say that there is an *undirected link* between v_1 and v_2 if there is a directed link from v_1 to v_2 and a directed link from v_2 to v_1. Let $S, R \in V$, $S = v_0$ and $R = v_{k+1}$, we say that $v_0, v_1, \ldots, v_k, v_{k+1}$ form a *directed (undirected) path* from (between) S to (and) R if there is a directed (undirected) link from (between) v_i to (and) v_{i+1} for each $0 \leq i \leq k$.

Neighbour networks, also defined in [FY95], are special multicast networks. The underlying graph of a neighbour network is an undirected graph $G(V, E)$, in which an element multicast by a node $v \in V$ will be received—simultaneously and privately—by all its neighbours, where a neighbour of v is a node $v' \in V$ such that $(v, v') \in E$.

It is easy to observe that a neighbour network graph can be used to model a special hypergraph in which all links are undirected, and vice versa. Therefore, given a neighbour network graph $G(V, E)$, we use $H_G(V, E_{H_G})$ as the hypergraph that graph G models. In this thesis, we mainly study RMT and SMT in the neighbour network graph $G(V, E)$, but we will use the results generalized from $H_G(V, E_{H_G})$.

2.2.3 Network Connectivity

In this thesis, we use a graph $G(V, E)$ to model a point-to-point network or a multicast communication neighbour network, and use $H(V, E_H)$ to denote a multicast communication hypergraph. Next, we define different kinds of network connectivities in different adversary models.

In the threshold (t-bounded adversary) model, we define the following network connectivities.

Definition 2.2.1. *Given a multicast graph $G(V, E)$ where $S, R \in V$, we say that S and R are* t-connected *if there are $n > t$ node-disjoint paths between S and R in G.*

Definition 2.2.2. (following [DW02]) *Given a multicast graph $G(V, E)$ where $S, R \in V$, we say that S and R are* $t_{neighbour}$-connected *if, after removing t nodes (excluding S and R) and their neighbours from G, there remains a path between S and R.*

Definition 2.2.3. (following [WD99]) *Given a multicast graph $G(V, E)$ where $S, R \in V$, we say that S and R are* weakly (n, t)-connected *if there are n node-disjoint paths $p_1, \ldots, p_n$ between S and R, and after removing t nodes (excluding S and R) and their neighbours from G, there remains a path p_i between S and R where $1 \leq i \leq n$.*

Definition 2.2.4. (following [FY95]) *Given a hypergraph $H(V, E_H)$ where $S, R \in V$, we say that S and R are* strongly (weakly) t_{hyper}-connected *if for any set $A \subseteq V \setminus \{S, R\}$ such that $|A| \leq t$, after removing all nodes in A and all hyperedges (v, V^*) such that $A \cap (\{v\} \cup V^*) \neq \emptyset$, there remains a directed (undirected) path from (between) S to (and) R.*

In the general adversary model (see Section 1.2), we define network connectivity as follows.

Definition 2.2.5. *Given a graph $G(V, E)$ where $S, R \in V$, let $A \subseteq V \setminus \{S, R\}$ be a set of nodes, if there is path p between S and R such that p passes through some nodes in A, then we say that A* cuts *p.*

Definition 2.2.6. (following [DWB05]) *Given a graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, if there exists a set $A \in \mathcal{A}$ such that A cuts all paths between S and R, then we say that S and R are* $\mathcal{A}$-separated*. We say that S and R are* $\mathcal{A}$-connected *if they are* not *$\mathcal{A}$-separated.*

Definition 2.2.7. *Let $k \geq 1$ and $\mathcal{A}$ be an adversary structure on a party set D, we define*

$$k\mathcal{A} = \{A_1 \cup \ldots \cup A_k | A_1, \ldots, A_k \in \mathcal{A}\}.$$

It is straightforward that $k\mathcal{A}$ is also an adversary structure on D. Let $D = V \setminus \{S, R\}$ in a graph $G(V, E)$, then when we say that S and R are $k\mathcal{A}$-connected, we mean that the union of any k sets in $\mathcal{A}$ cannot cut all the paths between S and R.

Regarding directed graphs, we give two special kinds of connectivities as follows.

Definition 2.2.8. *Given a directed graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, we say that S and R are* strongly $3\mathcal{A}$-directed-connected *if they are $2\mathcal{A}$-connected on the forward paths, and for any three sets $A_1, A_2, A_3 \in \mathcal{A}$, if $A_1 \cup A_2 \cup A_3$ cuts all the forward paths, then at most one of these three sets cuts all the feedback paths.*

Definition 2.2.9. *Given a directed graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, we say that S and R are* strongly $2\mathcal{A}$-directed-connected *if they are $\mathcal{A}$-connected on the forward paths, and $2\mathcal{A}$-connected with the union of all the forward and feedback paths in G.*

These special kinds of connectivities will be used when studying SMT in directed graphs.

2.3 Adversary Model

We consider an adversary who is characterized by an adversary structure, which is defined in Definition 1.2.1.

An adversary can be either *passive* or *active*. Assume that an adversary chooses a set of parties $A \in \mathcal{A}$ to control, where $\mathcal{A}$ is an adversary structure on $V \setminus \{S, R\}$ in a graph $G(V, E)$. A passive adversary only observes the traffic of the nodes in A. On the other hand, not only does an active adversary read the traffic through the nodes it controls, but it also decides whether to deny or to modify the transferred data, or whether to follow the protocol or not, on the nodes in A that it controls.

In this thesis we consider an adversary who can exhibit an active behaviour. Such an adversary has unlimited resources, and knows the complete protocol specification, message space and the complete structure of the graph.

In our work, an dynamic adversary, who can change the parties it controls from round to round, is not considered. Instead we only consider static adversaries. That is, before the protocol starts, the adversary chooses which set of parties $A \in \mathcal{A}$ to control, and the choice will not be changed until the end of the protocol.

2.4 Security Model

In this section we use a similar security model to that of [FW98].[1]

Given a graph $G(V, E)$ where $S, R \in V$, let Π be a message transmission protocol. The sender S starts with a message $m^S \in \mathbb{M}$ drawn with respect to a certain probability distribution. At the end of the protocol, the receiver R outputs a message $m^R \in \mathbb{M}$.

[1] We notice that some extended security models have been proposed in recent studies (e.g., in [KS09a, DESN10]), but in this thesis we only employ Franklin and Wright's model because it is a nice and simple measure of different security levels. Our results can be easily generalized in the extended models.

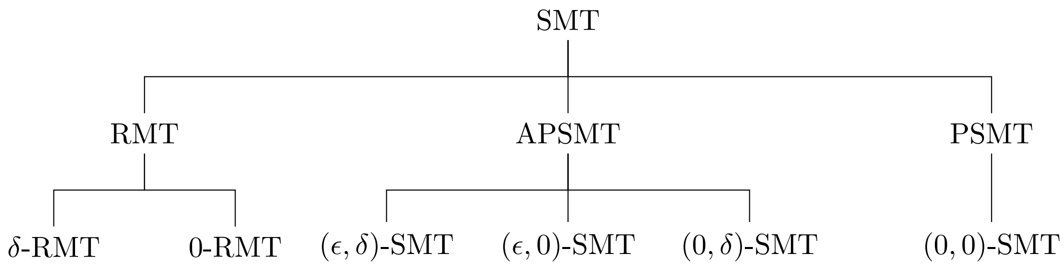


Figure 2.1: Extended problem set of SMT.

During the execution of the protocol Π, each node that participates in communication generates randomness's and performs local computation.

For any execution of the protocol Π, let *adv* be the adversary's view of the entire protocol, i.e., the behaviour of the faulty nodes, the initial state of the adversary, and the randomness's of the adversary during the execution. We write $adv(m, r)$ to denote the adversary's view when $m^S = m$ and when the randomness's generated by the adversary are r.

Privacy. The protocol Π is *ϵ-private* if, for any two messages $m_0, m_1 \in \mathbb{M}$ and any randomness's r, $\sum_c |\Pr[adv(m_0, r) = c] - \Pr[adv(m_1, r) = c]| \leq 2\epsilon$. The probabilities are taken over the randomness's of the honest parties, and the sum is over all possible values of the adversary's view.

Reliability. The protocol Π is *δ-reliable* if, with probability at least $1 - \delta$, R outputs $m^R = m^S$ at the end of the protocol. The probability is over the choice of m^S and the randomness's of all parties.

Security. The protocol Π is *(ϵ, δ)-secure* if it is ϵ-private and δ-reliable.

Therefore, using this notation, PSMT is written as $(0, 0)$-SMT. In the context of almost perfect security, both ϵ and δ should be made *negligible* in the security parameters. *Note* that in the rest of this thesis, ϵ and δ only appear when studying *almost perfect* security or reliability (i.e., $\epsilon > 0$ and $\delta > 0$), except when explicitly specified. With this security model, we extend the context of SMT as shown in Figure 2.1.

2.5 Communication Cost

The communication cost of an SMT protocol is calculated in communication complexity (CC) and round complexity (RC). CC is the worst case number of bits transmitted during the execution of the protocol. As we denoted in Section 2.1, field elements of length ρ are transmitted in our protocols. Thus if, for example, in a protocol $O(n)$ field elements are communicated, then the CC of this protocol is $O(n\rho)$. RC is the number of rounds taken by a protocol, where a round is a communication from S to R or vice versa.

In the previous studies of the threshold (t-bounded) model, Srinathan et al. showed in [SNR04] that the lower bound on the communication overhead is $\Omega(\frac{n}{n-2t})$ for 2-round PSMT, and Fitzi et al. showed in [FFGS07] that the lower bound for 1-round PSMT is $\Omega(\frac{n}{n-3t})$, where n is the number of node-disjoint paths between S and R. If we consider tight graphs, i.e., $n = 2t + 1$ for 2-round PSMT and $n = 3t + 1$ for 1-round PSMT, then the optimal CC is $O(n)$ for transmitting only one bit. This, however, is not possible in most message transmission protocols, which only enable security for large field elements. Thus if a message of size ℓ (bits) is transmitted, then the CC of an optimal protocol is $O(n\ell)$ (see, e.g., [SNR04, ACdH06, FFGS07]). As shown in Section 2.1, in our model, we let each message be a field element of size ρ (bits). In the case that a larger message of size $\ell\rho$ (bits) is transmitted, we model the transmission as ℓ messages each of size ρ are transmitted. Thus, throughout this thesis, we use ℓ to denote the number of messages (field elements) to be transmitted in a protocol, and hence the CC of an optimal protocol (in the threshold model) is $O(n\ell\rho)$.[2]

In general, a protocol is called *efficient* if its CC is polynomial in the size of the graph and its RC is constant. However, this notation has some limitations in the general adversary model. This is discussed further in Section 3.3.

2.6 Secret Sharing

Secret sharing schemes (SSSs) are extremely important in the studies of SMT. Given a set of n participants $D = \{1, \dots, n\}$, the idea of secret sharing is to divide a secret s into n pieces, where each piece is called a *share*, and assign a share to each participant. A *perfect* SSS enables any *qualified* subset (of D) of participants (i.e., any subset of participants that is by some law qualified to access the secret s) to reconstruct the secret s with the shares they hold, but does not reveal any information to any *unqualified* subset of participants. To enable perfect secret sharing, the size of a share must be larger or equal to the size of the secret [CSV93].

Note that if no unqualified subset becomes qualified by adopting a participant a, then the share of a is unimportant for any reconstruction of the secret. Thus an SSS does not need to assign a share to the unimportant participant a in the first place. Without loss of generality, we assume that all participants considered are important, and a share should be assigned to every one of them.

The qualified subsets are defined either by a threshold t or by an *access structure*

[2] There are some other measures of the communication cost of an SMT protocol. For example, *transmission rate* (see, e.g., [KS08]) and *transmission complexity* (see, e.g., [YD10]) are both used in the literature. These concepts are derived from the definition of the communication complexity (CC) and used for different purposes. Our CC model can explain our results well, so no other concepts will be employed in this thesis.

Γ. Based on the definitions of qualified subsets, we consider two kinds of SSSs: the threshold SSSs and the generalized SSSs. Next we discuss these two kinds of secret sharing in Section 2.6.1 and Section 2.6.2 respectively.

2.6.1 Threshold Secret Sharing

Regarding threshold secret sharing, we consider both *perfect* and *almost perfect* SSSs. A perfect SSS gives no advantage for the reconstruction of the secret to any set of less than $t+1$ participants, while an almost perfect SSS allows some set of at most t participants to reconstruct the secret with a small probability using the shares they hold.

Definition 2.6.1. *Let $\kappa < 1$ and $d \geq 1$, a $(t+1, n, \kappa)$-SSS is a probabilistic function $TS : \mathbb{F}^d \to \mathbb{F}^n$ such that for any secret $s \in \mathbb{F}$ and random vector $\mathbf{r} \in \mathbb{F}^{d-1}$, we have $TS(s, \mathbf{r}) = (s_1, \ldots, s_n)$. Let $1 \leq k \leq n$ and X be a variable induced by s, for any k entries $s_{i_1}, \ldots, s_{i_k}$ $(1 \leq i_1 < \ldots < i_k \leq n)$, a $(t+1, n, \kappa)$-SSS has the following two properties:*

property-1 $\Pr[X = s|s_{i_1}, \ldots, s_{i_k}] = 1$ *if $k \geq t+1$. That is, s can be reconstructed from any $k \geq t+1$ entries of $(s_1, \ldots, s_n)$ with probability 1.*

property-2 *For any secret $s' \in \mathbb{F}$, $\Pr[X = s'|s_{i_1}, \ldots, s_{i_k}] \leq \Pr[X = s'] + \kappa < 1$ if $k < t+1$. That is, s can be reconstructed from some $k < t+1$ entries with a probability less than 1.*

A perfect threshold SSS is thus a $(t+1, n, 0)$-SSS. A classic $(t+1, n, 0)$-SSS scheme has been designed by Shamir in [Sha79]. Shamir's SSS is based on polynomial interpolation. A Shamir's $(t+1, n, 0)$-SSS is constructed by using a polynomial $f(x)$ of degree at most t such that $f(x) = s + a_1x + \ldots + a_tx^t$ where $a_1, \ldots, a_t \in_R \mathbb{F}$, and the n shares are $(s_1, \ldots, s_n) = (f(1), \ldots, f(n))$. Independently, Blakley [Bla79] introduced another threshold SSS. Blakley's scheme is geometric in nature, as it divides the secret to n shares and each share defines an affine hyperplane. The secret can be reconstructed by finding the intersection of any $t+1$ hyperplanes. It is easy to observe that Blakley's SSS does not achieve perfect secrecy, because any t participants can get a line by intersecting their t hyperplanes, and they know that the secret is on this line, which may be a smaller space than $\mathbb{F}$. Therefore, Blakley's SSS is a $(t+1, n, \kappa)$-SSS with almost perfect secrecy. Shamir's SSS is more efficient than Blakley's, and more popular in application.

In Shamir's SSS, the size of the share is normally equal to the size of the secret.

2.6.2 Generalized Secret Sharing

In [ISN87], Ito et al. introduced the concept of the general *access structures.*

Definition 2.6.2. (following [ISN87, HM00]) *Given a participant set D, a monotone* access structure Γ *is a family of subsets $\Gamma \subseteq 2^D$ such that for any $A \in 2^D$, if $A \in \Gamma$ and $A \subseteq A'$, then $A' \in \Gamma$.*

Thus every set in Γ is qualified to access the secret. Without loss of generality, we assume that $\Gamma \neq \emptyset$.

In the generalized model, we consider only *perfect* SSSs. That is, given a set of participants $A \in 2^D$ and a secret $s \in \mathbb{F}$, a *generalized SSS* allows the participants in A to reconstruct the secret s with probability 1 if $A \in \Gamma$. However, if $A \notin \Gamma$, then the probability that the participants in A learn s is equal to the probability that they guess s with no knowledge of any share.[3]

Suppose that the parties in a network are the participants in D of an SSS, then an adversary structure can be seen as the complement of an access structure; i.e., $\mathcal{A} = 2^D \backslash \Gamma$. Therefore, the secrecy of an SSS means the privacy of an SMT protocol, if a message is shared among the parties.

The basic idea of the generalized SSS designed by Ito et al. [ISN87] is, for each set $A \in \Gamma$, to share the secret with Shamir's SSS among the participants in A. Since there are up to $O(2^n)$ sets in Γ, the size of the shares is exponentially large. Another generalized SSS was proposed by Benaloh and Leichter in [BL88]. Their scheme employs some carefully designed monotone functions that correspond to the properties of monotone access structures. Their scheme is slightly more efficient, but still requires the size of shares to be exponential in n. Many recent studies used linear secret sharing schemes (LSSSs) by applying monotone span programs (MSP) to share secrets among access structures (see, e.g., [Bei96]). This is studied in more detail in Section 3.2.

In general, the size of the largest share generated by a generalized SSS is larger than the size of the secret. There are a number of studies attempting to determine the size of the shares in the context of "information rate".[4] For example, in [Csi97], Csimaz constructed a special access structure such that if the size of the secret is ρ, then the lower bound on the size of the largest share is $\Omega(\frac{n\rho}{\log n})$. This lower bound is later corresponded by Blundo et al. [BSSV97], who showed a tight lower bound in their model. The result in this area will not be discussed in this thesis, but we refer to some studies in [Csi97, vD95, BSSV97]. However, to the best of our knowledge, there is no generalized SSS for *any* access structure with the size of the shares polynomial in n.

2.7 Error-Correcting Codes

In [MS81], McEliece and Sarwate argued that Shamir's SSS [Sha79] is very closely related to Reed-Solomon coding schemes [RS60]. In fact, Shamir's scheme corresponds to a special case of Reed-Solomon codes. This argument realizes the possibility of applying

[3] Due to the trivialness of the concept, a formal definition to the generalized SSS is not provided here. In Section 3.2.1, we give a formal definition of the generalized Linear SSS (LSSS) in Definition 3.2.2, which is more precise in concept.

[4] The information rate of an SSS is the ratio between the size of the secret and the largest share given to any participant. The studies of information rate often employ the connection between entropy and matroid (see e.g., [BD91, KOS$^+$93]). The findings on upper bound on the information rate result in the lower bound on the size of the shares.

error-correcting codes to SSSs.

We remark that a $(t+1, n, \kappa)$-SSS is a function $TS : \mathbb{F}^d \to \mathbb{F}^n$ $(d \geq 1)$ such that for any secret $s \in \mathbb{F}$ and random vector $\mathbf{r} \in \mathbb{F}^{d-1}$, we have $TS(s, \mathbf{r}) = (s_1, \ldots, s_n)$. The function can be seen as a coding function such that for each $(s, \mathbf{r}) \in \mathbb{F}^d$, $TS(s, \mathbf{r}) = (s_1, \ldots, s_n)$ is a codeword.

- We say that a $(t+1, n, \kappa)$-SSS can *detect* e errors if given any codeword $(s_1, \ldots, s_n)$ and any vector $(x_1, \ldots, x_n) \in \mathbb{F}^n$ such that $0 < |\{i : x_i \neq s_i, 1 \leq i \leq n\}| \leq e$, one can detect that $(x_1, \ldots, x_n)$ is not a codeword.

- We say that a $(t+1, n, \kappa)$-SSS can *correct* e errors if given any codeword $(s_1, \ldots, s_n)$ and any vector $(x_1, \ldots, x_n) \in \mathbb{F}^n$ such that $0 < |\{i : x_i \neq s_i, 1 \leq i \leq n\}| \leq e$, one can reconstruct the secret s from $(x_1, \ldots, x_n)$ with probability 1.

Reed-Solomon codes are maximum distance separable (MDS) codes, whose minimum Hamming distance[5] $d = n - b - 1$, where n is the length of the codewords and b is the dimension of the code. An MDS code with minimum Hamming distance d can detect $d - 1$ errors and correct $\lfloor \frac{d-1}{2} \rfloor$ errors [MS78]. As mentioned above, a Shamir's $(t+1, n, 0)$-SSS corresponds to a Reed-Solomon MDS code, where the length of the codewords is n and the dimension of the code is $b = t + 1$. Thus the minimum distance $d = n - (t+1) - 1 = n - t$. This implies that such a perfect $(t+1, n, 0)$-SSS can detect $n - t - 1$ errors and correct $\lfloor \frac{n-t-1}{2} \rfloor$ errors.

Berlekamp-Welch algorithm is a simple polynomial time decoding algorithm for Reed-Solomon codes. A nice description of this algorithm can be found in [HP06].

Later in Section 3.1, we examine the error-correcting capability of an almost perfect $(t+1, n, \kappa)$-SSS.

2.8 Authentication Codes

Authentication codes are constantly used in the protocols that achieve δ-reliability (see, e.g., [GMS74, RBO89, Rab94, FW98, WD99]).

Definition 2.8.1. *Let key* $= (a_0, a_1, \ldots, a_k) \in_R \mathbb{F}^{k+1}$ *be an authentication key of* $k+1$ *field elements, and let* $m \in \mathbb{F}$, *we define an authentication code*

$$\text{auth}(m; key) = a_0 + a_1 m + \ldots + a_k m^k.$$

Next we show that such an authentication code can authenticate k messages without giving away an advantage in forging the authentication key *key*.

[5] The Hamming distance between two codewords is the number of places where they differ.

Theorem 2.8.1. (following [WD99]) *Let* $key = (a_0, a_1, \ldots, a_k) \in_R \mathbb{F}^{k+1}$ *be an authentication key, and for* $1 \leq i \leq k$*, let* $m_i \in \mathbb{F}$ *and* $c_i = \text{auth}(m_i; key)$*, then for any* $a'_0, a'_1 \ldots, a'_k \in \mathbb{F}$*,*

$$\Pr[a_0 = a'_0|view] = \Pr[a_1 = a'_1|view] = \ldots = \Pr[a_k = a'_k|view] = \frac{1}{|\mathbb{F}|},$$

where $view = (m_1, c_1, \ldots, m_k, c_k)$*.*

Proof. According to Definition 2.8.1, we have the following:

$$\begin{bmatrix} 1 & m_1 & \cdots & m_1^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & m_k & \cdots & m_k^k \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} c_1 \\ \vdots \\ c_k \end{bmatrix}. \tag{2.1}$$

Since the coefficient matrix of eq. 2.1 is a $k \times (k+1)$ Vandermonde matrix, for each $0 \leq i \leq k$, no value of a_i can be ruled out. That is, given the values of $view = (m_1, c_1, \ldots, m_k, c_k)$, each a_i is equally likely to be any field element in $\mathbb{F}$. This concludes the proof. □

Due to Theorem 2.8.1, we have that a $key \in_R \mathbb{F}^{k+1}$ can be used to authenticate k elements without giving away an advantage in forging the key *key*.

2.9 Previous Results (in Detail)

In Section 1.3, we briefly surveyed the work in the area of SMT studies. Now after presenting our network, adversary and security models and related definitions and notations, we can discuss in more detail some previous results in this context.

2.9.1 On SMT in Point-to-Point Networks

First we present the previous results on SMT in point-to-point networks.

In the threshold model, the paths for communication between S and R must be *node-disjoint*, so an adversary who can control up to t nodes is not able to control more than t paths. In an undirected graph with n node-disjoint paths between S and R, 0-RMT or $(0,0)$-SMT is possible if and only if $n \geq 2t+1$ [DDWY93]. This condition is also the lower bound for δ-RMT and $(0,\delta)$-SMT, as shown in [FW98]. On the other hand, in a directed graph with n forward paths from S to R and u feedback paths from R to S, the necessary and sufficient condition for $(0,0)$-SMT is $n \geq \max\{3t+1-2u, 2t+1\}$ [DW02]. Different to the case in undirected graphs, the connectivity for both δ-RMT and $(0,\delta)$-SMT in a directed graph is weaker than that for $(0,0)$-SMT. Indeed, only $n \geq \max\{2t+1-u, t+1\}$ node-disjoint paths are required for δ-RMT or $(0,\delta)$-SMT in directed graphs [DW02].

Undirected Graphs			
SMT	RMT	δ-RMT	$n \geq 2t+1$ [FW98]
			N/A
		0-RMT	$n \geq 2t+1$ [DDWY93]
			$2\mathcal{A}$-connectivity [KGSR02]
	APSMT	(ϵ, δ)-SMT	N/A
			N/A
		$(\epsilon, 0)$-SMT	N/A
			N/A
		$(0, \delta)$-SMT	$n \geq 2t+1$ [FW98]
			N/A
	PSMT	$(0, 0)$-SMT	$n \geq 2t+1$ [DDWY93]
			$2\mathcal{A}$-connectivity [KGSR02]

Directed Graphs			
SMT	RMT	δ-RMT	$n \geq \max\{2t+1-u, t+1\}$ [DW02]
			N/A
		0-RMT	$n \geq 2t+1$ [DDWY93]
			$2\mathcal{A}$-connectivity on forward paths [DWB05]
	APSMT	(ϵ, δ)-SMT	N/A
			N/A
		$(\epsilon, 0)$-SMT	N/A
			N/A
		$(0, \delta)$-SMT	$n \geq \max\{2t+1-u, t+1\}$ [DW02]
			N/A
	PSMT	$(0, 0)$-SMT	$n \geq \max\{3t+1-2u, 2t+1\}$ [DW02]
			strong $3\mathcal{A}$-directed-connectivity [PSC$^+$07]

* For each security level (e.g., δ-RMT), the results are presented in two rows: the upper row indicates the result in the threshold model and the lower row indicates the result in the general adversary model.

Table 2.1: Network connectivities for SMT in point-to-point networks.

In the general adversary model, necessary and sufficient conditions for PSMT have been obtained in previous studies. In an undirected graph, 0-RMT and $(0, 0)$-SMT is possible if and only if S and R are $2\mathcal{A}$-connected [KGSR02]. In a directed graph, 0-RMT is possible if and only if S and R are $2\mathcal{A}$-connected on the forward paths [DWB05], and $(0, 0)$-SMT is possible if and only if S and R are strongly $3\mathcal{A}$-directed-connected (see Definition 2.2.8) [PSC$^+$07]. Unlike the results in the threshold model, the conditions for SMT in the general adversary model do not require node-disjoint paths.[6] The communication on a network graph is normally via the so-called critical paths (see our model in Section 3.3). In [KGSR02], an algorithm for identifying critical paths from all paths in a graph has been proposed. A sketch of the algorithm is shown in Appendix A.1.

In summary, the previous results regarding the point-to-point network connectivities are shown in Table 2.1, with the N/A (not available) parts to be filled in by this thesis.

[6] In [PSC$^+$07], the condition for $(0, 0)$-SMT requires node-disjoint paths. This requirement is clearly unnecessary and incorrect.

		Privacy		
		None (1)	Almost perfect (ϵ)	Perfect (0)
Reliability	Almost perfect (δ)	$n > t$	$n > t$	$n > t$
	Perfect (0)	$n > 2t$	$n > 2t$	$n > 2t$

Table 2.2: Network connectivities for SMT on multicast channels [FW98].

As we mentioned in Section 1.3, there are a series of studies which considered the communication costs of the PSMT protocols in the threshold model; e.g., [SAA96, SNR04, ACdH06, PCSR06, FFGS07, KS08]. The best known result has been obtained in [KS08], in which a 2-round PSMT protocol is proposed, and the CC of the protocol is $O(n\ell\rho)$ while transmitting ℓ (ℓ is polynomial in n) messages. To obtain this result, some techniques given in [SNR04], including *randomness extractor* (see Appendix A.2) and *advanced reliable transmission* (see Appendix A.3), have been used, and a new idea of generating *pseudo-basis* and *pseudo-dimension* (see Appendix A.6) has been introduced in [KS08]. Some of these techniques are employed and generalized in our study of PSMT protocols in the general adversary model.

In the literature, no specific result has been found regarding the communication costs of PSMT protocols in the general adversary model. In [KGSR02], the proposed PSMT protocol, which is executed in a $2\mathcal{A}$-connected undirected graph, has an RC $O(n)$ and a CC $O(hn^2\rho)$, where h is the total size of the shares constructed by a generalized SSS. In [DWB05], a 1-round PSMT protocol, requiring S and R to be $3\mathcal{A}$-connected, has been given with a CC $O(|\mathcal{A}|n\rho)$. In [PSC$^+$07], a PSMT protocol with strong $3\mathcal{A}$-directed-connectivity is proposed by combining a quasi-polynomial (in $|\mathcal{A}|$) number of sub-protocols and error-correcting codes (see Appendix A.5). Thus both the RC and CC of their protocol are quasi-polynomial in the size of the adversary structure, which is generally exponential in the size of the graph. The comparisons on RC and CC between our results and the results mentioned above are presented in Section 4.3.1.

2.9.2 On SMT in Multicast Networks

Concerning multicast communication on neighbour networks, if the adversary is a passive eavesdropper, then as proven in [FY95], the weak t_{hyper}-connectivity (see Definition 2.2.4) is necessary for 0-private communication. In the presence of a Byzantine (active) adversary, the results in the threshold model have been given in [FW98]. These results, requiring n node-disjoint and neighbour-disjoint paths, can be summarized in Table 2.2. In [FW98], Franklin and Wright showed an efficient $(0, \delta)$-SMT protocol for $n > \lceil \frac{3t}{2} \rceil$, and asked an open question on whether there exists an efficient $(0, \delta)$-SMT protocol for $t < n \leq \lceil \frac{3t}{2} \rceil$. Wang and Desmedt answered this question in [WD99], by showing an efficient $(0, \delta)$-SMT protocol that employs the properties of the authentication codes (see Section 2.8).

Notably the most significant finding in the multicast neighbour network setting is

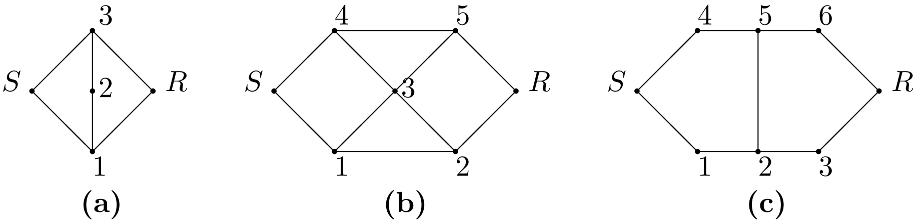


Figure 2.2: Private and reliable connectivity.

that the connectivity required for δ-RMT, and hence $(0, \delta)$-SMT, is unexpectedly weaker than that required in point-to-point networks (i.e., $n > t$ vs. $n > 2t$).

As discussed in Section 1.3, the results of [FW98] are obtained under a strong assumption that the n paths between S and R are neighbour-disjoint. The necessary and sufficient conditions for SMT in a more general multicast graph setting remain unknown. In [WD99], Wang and Desmedt claimed that the weak (n, t)-connectivity (see Definition 2.2.3) is sufficient for $(0, \delta)$-SMT in multicast graphs.[7] In [DW02], Desmedt and Wang extended this study. Using examples, they showed that the following implications are strict (see related definitions in Section 2.2.3):

$$\begin{aligned}\text{weak } (n,t)\text{-connectivity } &\Rightarrow t_{neighbour}\text{-connectivity}\\ &\Rightarrow \text{weak } t_{hyper}\text{-connectivity } \Rightarrow t\text{-connectivity.}\end{aligned}$$

In addition, they gave the following results regarding SMT in multicast graphs [DW02]:

- The weak (n, t)-connectivity is not necessary for $(0, \delta)$-SMT in multicast graphs. E.g., in Figure 2.2(a), S and R are not weakly $(2, 1)$-connected, but $(0, \delta)$-SMT is possible (see their protocol in [WD08]).

- As proven in [FY95], the weak t_{hyper}-connectivity is necessary for 0-private communication, and hence $(0, \delta)$-SMT. However, the weak t_{hyper}-connectivity is not necessary for δ-RMT. E.g., in Figure 2.2(b), S and R are not weakly 1_{hyper}-connected, but δ-RMT is possible.

- They conjectured that the weak t_{hyper}-connectivity is not sufficient for $(0, \delta)$-SMT. E.g., in Figure 2.2(c), S and R are weakly 1_{hyper}-connected, but they conjectured that there is no $(0, \delta)$-SMT against a 1-bounded adversary.

Therefore, regarding SMT in the general multicast graph setting, there is no previous result that gives the necessary and sufficient conditions for any level of security. Indeed, if we draw a similar table as Table 2.1 for SMT in multicast graphs, then all the result fields will be N/A. In this thesis we completely solve this problem, and our findings will be shown in Chapter 5.

[7] Wang and Desmedt believed that the sufficiency of this condition is straightforward, so in their papers [WD99, WD01], no proper proof was given. We see this claim as a conjecture, and this matter will be discussed in more detail in Section 5.4.

Chapter 3

Ideas and Observations

Secret sharing schemes (SSSs) and error-correcting codes play critically important roles in the study of secure message transmission (SMT). Indeed, not only can they help design SMT protocols, but they can also be used to determine the minimal network connectivity for a certain level of security. One of the goals of this thesis is to determine the necessary condition for ϵ-private message transmission. We can achieve this goal by examining the error-correcting capability of a $(t+1, n, \kappa)$-SSS (See Definition 2.6.1).

Another goal of this thesis is to reduce the communication costs of the previous $(0,0)$-SMT protocols. To design efficient protocols, we regard monotone access structures as linear codes, whose properties can then be employed for efficient $(0,0)$-SMT protocol construction. Our result uses a generalization of Kurosawa and Suzuki's idea of *pseudo-basis* and *pseudo-dimension* presented in [KS08] (see Appendix A.6).

As discussed in the previous chapters, SMT in the general adversary model does not require node-disjoint paths between S and R. Therefore, a basic idea of using *critical paths* was introduced by Kumar et al. in [KGSR02] (see Appendix A.1). We show our observation on the properties of the critical paths, and discuss how to determine the communication complexity (CC) of SMT protocols using critical paths.

In this chapter we present the above ideas and observations in three respective sections. Some results in this chapter has been published in [YD09, YD10].

3.1 $(t+1, n, \kappa)$-SSS and Error-Correcting

In Section 2.7, we denoted the terms error-detecting and error-correcting for $(t+1, n, \kappa)$-SSS, and showed that there exists a perfect $(t+1, n, 0)$-SSS can detect $n-t-1$ errors and correct $\lfloor \frac{n-t-1}{2} \rfloor$ errors. In the following, using exhaustive search, we show that an almost perfect $(t+1, n, \kappa)$-SSS can do just the same.

Lemma 3.1.1. *Let* $(s, \mathbf{r}) \in \mathbb{F}^d$ *(*$d \geq 1$*) and* TS *be a* $(t+1, n, \kappa)$*-SSS such that* $TS(s, \mathbf{r}) = (s_1, \ldots, s_n)$*. For any* $t+1$ *entries* $s_{i_1}, \ldots, s_{i_{t+1}}$ *(*$1 \leq i_1 < \ldots < i_{t+1} \leq n$*), given* $TS(s', \mathbf{r}') = (s'_1, \ldots, s'_n)$ *for any* $s' \neq s$ *and any* $\mathbf{r}' \in \mathbb{F}^{d-1}$*, we have* $(s'_{i_1}, \ldots, s'_{i_{t+1}}) \neq$

$(s_{i_1}, \ldots, s_{i_{t+1}})$.

Proof. Due to property-1 of Definition 2.6.1, we have that $\Pr[X = s | s_{i_1}, \ldots, s_{i_{t+1}}] = 1$ and $\Pr[X = s' | s'_{i_1}, \ldots, s'_{i_{t+1}}] = 1$. Since $s' \neq s$, we have $\Pr[X = s | s'_{i_1}, \ldots, s'_{i_{t+1}}] = 0$. This implies that $(s'_{i_1}, \ldots, s'_{i_{t+1}}) \neq (s_{i_1}, \ldots, s_{i_{t+1}})$. □

Lemma 3.1.2. *Let $(s, \mathbf{r}) \in \mathbb{F}^d$ $(d \geq 1)$ and TS be a $(t+1, n, \kappa)$-SSS such that $TS(s, \mathbf{r}) = (s_1, \ldots, s_n)$. For any t entries $s_{i_1}, \ldots, s_{i_t}$ $(1 \leq i_1 < \ldots < i_t \leq n)$, there exists a codeword $TS(s', \mathbf{r}') = (s'_1, \ldots, s'_n)$ where $s' \neq s$ and $\mathbf{r}' \in \mathbb{F}^{d-1}$ such that $(s'_{i_1}, \ldots, s'_{i_t}) = (s_{i_1}, \ldots, s_{i_t})$.*

Proof. Assume that such codeword $TS(s', \mathbf{r}')$ does not exist. That is, only s can be reconstructed from these t entries $s_{i_1}, \ldots, s_{i_t}$; i.e., $\Pr[X = s | s_{i_1}, \ldots, s_{i_t}] = 1$. This violates property-2 of Definition 2.6.1, and hence we have a contradiction. □

Theorem 3.1.1. *A $(t+1, n, \kappa)$-SSS can detect $n - t - 1$ errors, but no more.*

Proof. Let $(s, \mathbf{r}) \in \mathbb{F}^d$ $(d \geq 1)$ and TS be a $(t+1, n, \kappa)$-SSS such that $TS(s, \mathbf{r}) = (s_1, \ldots, s_n)$. As discussed in Section 2.7, we regard $(s_1, \ldots, s_n)$ as a codeword. First we show that if there is a vector $\mathbf{x} = (x_1, \ldots, x_n)$ such that $|\{i : x_i \neq s_i, 1 \leq i \leq n\}| = e$ and $0 < e \leq n - t - 1$, then one can detect that $\mathbf{x}$ is not a codeword. Since $n - e \geq n-(n-t-1) = t+1$, there are at least $t+1$ entries $x_{i_1}, \ldots, x_{i_{t+1}}$ $(1 \leq i_1 < \ldots < i_{t+1} \leq n)$ such that $(x_{i_1}, \ldots, x_{i_{t+1}}) = (s_{i_1}, \ldots, s_{i_{t+1}})$. Due to Lemma 3.1.1, $x_{i_1}, \ldots, x_{i_{t+1}}$ can *only* be a part of the codeword $TS(s, \mathbf{r})$ for some $\mathbf{r} \in \mathbb{F}^{d-1}$. Since the e errors are not a part of this codeword, it is easy to detect that $\mathbf{x}$ is not a codeword.

Next we show that if $e > n - t - 1$, then the vector $\mathbf{x}$ can also be a codeword; i.e., $\mathbf{x} = TS(s', \mathbf{r}')$ where $s' \neq s$ and $\mathbf{r}' \in \mathbb{F}^{d-1}$. Since in this case $n-e < n-(n-t-1) = t+1$, there are at most t entries $x_{i_1}, \ldots, x_{i_t}$ $(1 \leq i_1 < \ldots < i_t \leq n)$ such that $(x_{i_1}, \ldots, x_{i_t}) = (s_{i_1}, \ldots, s_{i_t})$. Due to Lemma 3.1.2, there exists a secret $s' \neq s$ such that these $n - e$ entries are a part of the codeword $TS(s', \mathbf{r}')$, and it is possible that the e errors are also a part of this codeword. Thus $\mathbf{x}$ can be a codeword, and hence one cannot detect $e > n - k - 1$ errors. □

Theorem 3.1.2. *A $(t+1, n, \kappa)$-SSS can correct $\lfloor \frac{n-t-1}{2} \rfloor$ errors, but no more.*

Proof. Let $(s, \mathbf{r}) \in \mathbb{F}^d$ $(d \geq 1)$ and TS be a $(t+1, n, \kappa)$-SSS such that $TS(s, \mathbf{r}) = (s_1, \ldots, s_n)$. As discussed in Section 2.7, we regard $(s_1, \ldots, s_n)$ as a codeword. First we show that if there is a vector $\mathbf{x} = (x_1, \ldots, x_n)$ such that $|\{i : x_i \neq s_i, 1 \leq i \leq n\}| = e$ where $0 < e \leq \lfloor \frac{n-t-1}{2} \rfloor$, then one can reconstruct the secret s from $\mathbf{x}$. To correct e errors, one selects $n - e$ entries from $\mathbf{x}$ and puts them into a new $(n - e)$-vector $\mathbf{y}$. In this way, we can regard $\mathbf{y}$ as a codeword of a $(t+1, n-e, \kappa)$-SSS that shares s with at most e errors. Due to Theorem 3.1.1, a $(t+1, n-e, \kappa)$-SSS can detect

$$n - e - t - 1 \geq n - \lfloor \frac{n-t-1}{2} \rfloor - t - 1 \geq \lfloor \frac{n-t-1}{2} \rfloor \geq e$$

errors. With at most e errors in $\mathbf{y}$, one can detect if $\mathbf{y}$ is a codeword. If $\mathbf{y}$ is not a codeword, then one uses exhaustive search until it finds a $\mathbf{y}$ that is a (error-free) codeword, and then the secret s can be reconstructed from $\mathbf{y}$.

Next we show that if $e > \lfloor \frac{n-t-1}{2} \rfloor$, then one cannot correct e errors and reconstruct s from $\mathbf{x}$. We view $\mathbf{x}$ as it consists of three parts: (1) t error-free entries, (2) e errors and (3) $b = n - t - e$ other error-free entries. Let $e = \lfloor \frac{n-t-1}{2} \rfloor + a$ $(a \geq 1)$, we have the following:

$$\begin{aligned} b = n - t - e = n - t - (\lfloor \frac{n-t-1}{2} \rfloor + a) \\ &\leq 2 \times \lfloor \frac{n-t-1}{2} \rfloor + 2a - (\lfloor \frac{n-t-1}{2} \rfloor + a) \\ &= \lfloor \frac{n-t-1}{2} \rfloor + a \\ &= e. \end{aligned}$$

That is, $b \leq e$. Due to Lemma 3.1.2, the t error-free entries are a part of the codeword $TS(s, \mathbf{r})$ as well as $TS(s', \mathbf{r}')$ for some $s' \neq s$. It is possible that the e errors are also a part of the codeword $TS(s', \mathbf{r}')$. Therefore, those $t + b$ error-free entries are a part of $TS(s, \mathbf{r})$, and those $t + e$ entries are a part of $TS(s', \mathbf{r}')$. Since $b \leq e$, one cannot distinguish whether the secret s is shared and the e entries are errors, or the secret s' is shared and the b entries are errors. Thus one cannot reconstruct s with probability 1. □

Therefore, we have shown that an almost perfect $(t + 1, n, \kappa)$-SSS can detect and correct exactly same numbers of errors as a perfect $(t+1, n, 0)$-SSS can. This result can be used to determine the minimal network connectivity for ϵ-private communication. We discuss this further in Section 4.2.

3.2 Generalized Secret Sharing and Linear Codes

In this section, we first present the definition and construction of a generalized linear secret sharing scheme (LSSS) using a monotone span program (see Section 3.2.1), and then we propose a new generalized linear code (see Section 3.2.2) for the purpose of error-correcting and also for the purpose of defining pseudo-basis and pseudo-dimension (see Section 3.2.3). This follows the idea of Kurosawa and Suzuki in the threshold model [KS08].

3.2.1 Constructing an LSSS

Given a set of n participants $D = \{1, \ldots, n\}$, a generalized LSSS can be constructed to share a secret s among the participants, in such a way that any set of participants in an access structure Γ (see Definition 2.6.2) can reconstruct the secret, but any other set cannot.

First, it is well-known that monotone span programs are essentially equivalent to LSSSs (see [KW93, CDM00]).

Definition 3.2.1. (following [KW93]) *A* monotone span program *is a triple* $(\mathbb{F}, M, \psi)$*, where* $\mathbb{F}$ *is a finite field,* M *is an* $h \times d$ *matrix* $(h \geq d)$*, and* $\psi : \{1, \ldots, h\} \rightarrow \{1, \ldots, n\}$ *is a surjective function that assigns a number of rows in* M *to each participant in* D*.*

For later use, we only allow each row of M to be assigned to a unique participant; i.e., if $\psi(i) = j$, then $\psi(i) \neq j'$ for any $j' \neq j$. This is easy to achieve by duplicating the rows that are assigned to multiple participants. Thus h can indicate the total number of shares distributed.

Similar to Shamir's scheme, our construction assumes that $\mathbb{F}$ is sufficiently large. Now with $(\mathbb{F}, M, \psi)$, one can share a secret $s \in \mathbb{F}$ using an LSSS.

Definition 3.2.2. *Given a monotone span program* $(\mathbb{F}, M, \psi)$*, a secret* $s \in \mathbb{F}$ *and a random vector* $\mathbf{r} \in \mathbb{F}^{d-1}$*, we regard* $LS : \mathbb{F}^d \rightarrow \mathbb{F}^h$ *as a function such that (*T *denotes transpose)*

$$LS(s, \mathbf{r}) = \left(M \times (s, \mathbf{r})^T\right)^T = (s_1, \ldots, s_h),$$

where $s_1, \ldots, s_h$ *are the* h *shares generated by the LSSS, and they are assigned to the* n *participants by* ψ*. For any set* $A \in \Gamma$*, let the* t *shares* $s_{i_1}, \ldots, s_{i_t}$ $(1 \leq i_1 < \ldots < i_t \leq h)$ *be all the shares assigned to the participants in* A *(i.e.,* $\psi(i_1, \ldots, i_t) = A$*), and let* X *be a random variable induced by* s*, the LSSS must satisfy the following conditions:*

Secrecy: *let* s' *be any secret,* $\Pr[X = s' | s_{i_1}, \ldots, s_{i_t}] = \Pr[X = s']$ *if* $\psi(i_1, \ldots, i_t) \notin \Gamma$*;*

Reconstruction: $\Pr[X = s | s_{i_1}, \ldots, s_{i_t}] = 1$ *if* $\psi(i_1, \ldots, i_t) \in \Gamma$*.*

Apparently, if $\psi(i_1, \ldots, i_t) \in \Gamma$, then in the linear span of the $i_1, \ldots, i_t$-th rows of M, the *target vector* $tar = (1, 0, \ldots, 0)$ must exist [KW93]. This is required to satisfy the reconstruction condition.

As discussed in Section 2.6.2, in the context of the information rate, the size of the shares has been studied in literature (e.g., [Csi97, vD95, BSSV97]). However, to the best of our knowledge, there is no result regarding the tight upper bound on the total size of the shares, which is h in our LSSS. In fact, we do not know whether for any access structure, there exists an LSSS with size h polynomial in n. However, we can have an exponential size LSSS, which we call the *Worst Case LSSS*, as follows. This LSSS is derived from the generalized SSS proposed in [Mau06].

Given a set of n participants D and a monotone access structure Γ on D, we let $\mathcal{A} = 2^D \setminus \Gamma$. The worst case LSSS is defined by a monotone span program $(\mathbb{F}, M_{h \times d}, \psi)$ such that $d = |\mathcal{A}|$ and $h = O(dn)$. h is thus exponential in n because in general $d = |\mathcal{A}| = O(2^n)$.

Worst Case LSSS

Let $\Delta = \{D \setminus A | A \in \mathcal{A}\}$ and $d = |\Delta| = |\mathcal{A}|$. Construct a $d \times d$ matrix M^V, which is an identity matrix except all entries in the first row are 1.

Let $\Delta = \{B_1, \ldots, B_d\}$, then for each $1 \leq i \leq d$, construct a $|B_i| \times d$ matrix M_i such that each row of M_i is a duplication of the i-th row of M^V. Let $h = \sum_{i=1}^{d} |B_i|$, construct an $h \times d$ matrix M that is filled by $M_1, \ldots, M_d$ from top to bottom.

The function ψ assigns the rows in M to each participant in such a manner that if a participant is in $B_i \in \Delta$ $(1 \leq i \leq d)$, then ψ assigns a row of M_i to this participant. **End.**

Theorem 3.2.1. *The Worst Case LSSS is a generalized linear secret sharing scheme.*

Proof. We prove that the Worst Case LSSS satisfies the *Secrecy* and *Reconstruction* conditions of Definition 3.2.2.

There are in total h shares distributed, but only d distinct shares. Let $s_1, \ldots, s_d$ be the distinct shares of a secret s, then it is straightforward that s can only be reconstructed with *all* the d shares, because all rows in the matrix M^V are linearly independent.

First, for each set $A \notin \Gamma$, we have $A \in \mathcal{A}$ and there is a share s_i assigned to the set $B_i = D \setminus A$. Thus the participants in A cannot learn s_i and hence cannot reconstruct s. Thus the Secrecy condition is satisfied. Next, for contradiction, we assume that there exists a set $A' \in \Gamma$ such that the participants in A' cannot learn a share s_i. Let $A \in \mathcal{A}$ and $B_i = D \setminus A$, then A' does not contain any participant in B_i, which means $A' \subseteq A \Rightarrow A' \in \mathcal{A}$, and hence we have a contradiction. Thus any set in Γ can learn all the d shares to reconstruct the secret s, which means that the Reconstruction condition is satisfied. □

3.2.2 Linear Codes

Now we have an LSSS defined by $(\mathbb{F}, M_{h \times d}, \psi)$. We denote k as the rank of M, thus $k \leq d$. In the rest of the section, *we let the first k rows of M be linearly independent.*[1] Thus $\psi(1, \ldots, k) \in \Gamma$. Indeed, because otherwise $\psi(1, \ldots, k) \in \mathcal{A}$ and the participants in $\psi(1, \ldots, k)$ can then recover all the other shares using linear combinations, which contradicts the Secrecy condition of Definition 3.2.2. Next we define a generalized linear code.

Definition 3.2.3. *A* linear code *C is defined by a $k \times h$ generating matrix G in standard form $G = (I_k | N)$ [MS78], where I_k denotes the $k \times k$ identity matrix and N is a $k \times (h-k)$ matrix.*

[1] For example, in our worst case LSSS, it is straightforward that $k = d$. One can easily adjust the matrix M to make the first d rows be the rows of M^V, and hence the first d rows are linearly independent.

The codewords of code C are determined by an encoding *function $EC : \mathbb{F}^k \rightarrow \mathbb{F}^h$ such that given a k-vector $\mathbf{r} \in \mathbb{F}^k$,*

$$EC(\mathbf{r}) = \mathbf{r} \times G = \mathbf{c},$$

where $\mathbf{c}$ is an h-vector, as a codeword of C, and denoted $\mathbf{c} \in C$.

Evidently, the linear code C has $|\mathbb{F}|^k$ codewords, each of which is generated for a k-vector $\mathbf{r}$.

We link an LSSS with a linear code as follows. In the rest of this section, *we let M_k be a $k \times d$ matrix that consists of the first k rows of M*, so the rank of M_k is k. We construct the generating matrix G in such a manner that the i-th column of G, which we call col_i, has the following property:

$$(col_i)^T \times M_k = row_i, \tag{3.1}$$

where row_i is the i-th row of M. This is possible because the rank of M is k, thus each row_i is in the linear span of the first k rows of M (M_k). Therefore, the set of shares $\{LS(s, \mathbf{r}) | s \in \mathbb{F}, \mathbf{r} \in \mathbb{F}^{d-1}\}$ consists of the codewords of a linear code, because for any $s \in \mathbb{F}, \mathbf{r} \in \mathbb{F}^{d-1}$, we have

$$LS(s, \mathbf{r}) = (s_1, \ldots, s_h) = EC(s_1, \ldots, s_k) \in C.$$

Definition 3.2.4. *Let $\mathbf{k}$ be a k-vector such that $\mathbf{k} \times M_k = tar$, where $tar = (1, 0, \ldots, 0) \in \mathbb{F}^d$ is the target vector,*[2] *and let $\mathbf{r} \in \mathbb{F}^k$. We define a* decoding *function $DC : \mathbb{F}^k \rightarrow \mathbb{F}$ such that $DC(\mathbf{r}) = \mathbf{r} \times \mathbf{k}^T$. We denote the output of the function, $r = DC(\mathbf{r})$, as* the information of the codeword *$\mathbf{c} = EC(\mathbf{r})$.*

Lemma 3.2.1. *Given any codeword $\mathbf{c} = (c_1, \ldots, c_h) = EC(\mathbf{r}) \in C$, for any t entries $c_{i_1}, \ldots, c_{i_t}$ $(1 \leq i_1 < \ldots < i_t \leq h)$ which are all the entries assigned to a set of participants, one can decode the information of $\mathbf{c}$ with these t entries if and only if $\psi(i_1, \ldots, i_t) \in \Gamma$.*

Proof. Let $\mathbf{k}$ be a k-vector such that the information of $\mathbf{c}$ is $r = DC(\mathbf{r}) = \mathbf{r} \times \mathbf{k}^T$. From Definition 3.2.4, we have $\mathbf{k} \times M_k = tar$ where tar is the target vector. Remark that C is defined by the generating matrix G, which is derived from the matrix M of the LSSS. Let Λ be a $k \times t$ matrix such that for each $1 \leq j \leq t$, the j-th column of Λ is the i_j-th column of G, then due to eq. 3.1, we have

$$\begin{bmatrix} row_{i_1} \\ \vdots \\ row_{i_t} \end{bmatrix} = \Lambda^T \times M_k, \tag{3.2}$$

[2] Because $\psi(1, \ldots, k) \in \Gamma$, as shown before, the target vector must be in the linear span of the first k rows of M. Thus $\mathbf{k}$ must exist.

where for each $1 \leq j \leq t$, row_{i_j} is the i_j-th row of M.

First, we show that if $\psi(i_1, \dots, i_t) \notin \Gamma$, then one cannot decode r with $c_{i_1}, \dots, c_{i_t}$. Assume the opposite; i.e., r can be decoded with $c_{i_1}, \dots, c_{i_t}$. Since $r = \mathbf{r} \times \mathbf{k}^T$ and $(c_{i_1}, \dots, c_{i_t}) = \mathbf{r} \times \Lambda$, the possibility that r can be decoded with $c_{i_1}, \dots, c_{i_t}$ means that the column vector $\mathbf{k}^T$ is in the linear span of the columns of Λ. Indeed, there must exist a $\mathbf{t}^T$ such that $\mathbf{k}^T = \Lambda \times \mathbf{t}^T$, so that

$$r = \mathbf{r} \times \mathbf{k}^T = \mathbf{r} \times \Lambda \times \mathbf{t}^T = (c_{i_1}, \dots, c_{i_t}) \times \mathbf{t}^T.$$

Since $\mathbf{k}^T = \Lambda \times \mathbf{t}^T \Rightarrow \mathbf{k} = \mathbf{t} \times \Lambda^T$, by multiplying $\mathbf{t}$ by both sides of Eq. 3.2, we have

$$\mathbf{t} \times \begin{bmatrix} row_{i_1} \\ \vdots \\ row_{i_t} \end{bmatrix} = \mathbf{t} \times \Lambda^T \times M_k = \mathbf{k} \times M_k = tar.$$

This means that the target vector tar is in the linear span of the rows assigned to the participants in $\psi(i_1, \dots, i_t) \notin \Gamma$, which is not allowed in our LSSS due to the Secrecy condition. This contradiction proves the “only if” direction of our condition.

Next, if $\psi(i_1, \dots, i_t) \in \Gamma$, then using the reverse of the above proof and the Reconstruction condition of the LSSS, we can easily prove that one can decode r with $c_{i_1}, \dots, c_{i_t}$. □

Given that $\mathbf{c} = (c_1, \dots, c_h)$ is a codeword at the encoding end and $\mathbf{x} = (x_1, \dots, x_h)$ is the input at the decoding end, because of the channel noise, it is possible that $\mathbf{x} \neq \mathbf{c}$. We denote $\mathbf{e} = (e_1, \dots, e_h)$ as an *error vector* such that $\mathbf{e} = \mathbf{x} - \mathbf{c}$. Normally we have the following assumption: let $E = \{i | e_i \neq 0\}$ be an *error locator*, we always have $\psi(E) \in \mathcal{A}$, where $\mathcal{A} = 2^D \setminus \Gamma$ is an adversary structure. That is, the errors in a codeword are caused by a set in the adversary structure. Therefore, we have the following lemma:

Lemma 3.2.2. *Given a set of participants D, let Γ and $\mathcal{A}$ be an access structure and an adversary structure on D respectively, C be a generalized linear code, $\mathbf{c} \in C$ be a codeword at the encoding end and $\mathbf{x}$ be a vector at the decoding end:*

- *the decoder can detect that $\mathbf{x}$ is not a codeword if and only if $D \notin 2\mathcal{A}$;*
- *the decoder can decode the information of $\mathbf{c}$ from $\mathbf{x}$ if and only if $D \notin 3\mathcal{A}$.*[3]

It is easy to prove Lemma 3.2.2 using a similar exhaustive search technique as that of Section 3.1. Next, we generalize Kurosawa and Suzuki's idea of pseudo-basis and pseudo-dimension [KS08] (see also Appendix A.6) under the condition $D \notin 2\mathcal{A}$.

[3] $D \notin 2\mathcal{A}$ means $D \notin \{A_1 \cup A_2 | A_1, A_2 \in \mathcal{A}\}$ and $D \notin 3\mathcal{A}$ means $D \notin \{A_1 \cup A_2 \cup A_3 | A_1, A_2, A_3 \in \mathcal{A}\}$. See Definition 2.2.7.

3.2.3 Pseudo-Basis and Pseudo-Dimension

At Eurocrypt '08, Kurosawa and Suzuki initiated the idea of pseudo-basis and pseudo-dimension in the threshold model with multiple codewords. A generalization of the pseudo-basis and pseudo-dimension is possible if $D \notin 2\mathcal{A}$ (corresponding to $n \geq 2t+1$ in the threshold model), thus *we assume that $D \notin 2\mathcal{A}$ in this section.*

We let ψ^{-1} be the inverse function of ψ. That is, let $A \subseteq D$, then $\psi^{-1}(A)$ returns all the locations of the entries in a codeword that are assigned to the participants in A by ψ.

Definition 3.2.5. *Let $A \subseteq D$, we define $|A|$ as the* size *of A and $|\psi^{-1}(A)|$ as the* weight *of A. We denote $sz^{\mathcal{A}} = \max\{\text{size of } A | A \in \mathcal{A}\}$ and $wt^{\mathcal{A}} = \max\{\text{weight of } A | A \in \mathcal{A}\}$.*

Apparently, $sz^{\mathcal{A}} = O(n)$ and $wt^{\mathcal{A}} = O(h)$. The idea of the generalization is as follows. The encoder sends w codewords $\mathbf{c}_1, \ldots, \mathbf{c}_w$, and the decoder receives w h-vectors $\mathbf{x}_1, \ldots, \mathbf{x}_w$ such that for each $1 \leq i \leq w$, $\mathbf{x}_i = \mathbf{c}_i + \mathbf{e}_i$ where $\mathbf{e}_i = (e_{i1}, \ldots, e_{ih})$ is an error vector. For each $\mathbf{e}_i$, let $E_i = \{j | e_{ij} \neq 0\}$ be an error locator, then E_i has the following two properties: (1) $|E_i| \leq wt^{\mathcal{A}}$ and (2) $\psi(E_i) \in \mathcal{A}$ and hence $|\psi(E_i)| \leq sz^{\mathcal{A}}$. We assume that $\psi(\bigcup_{i=1}^{w} E_i) \in \mathcal{A}$. That is, the errors in all codewords are caused by the same set in $\mathcal{A}$. Now we give our Pseudo-Basis Construction Scheme as follows.

Pseudo-Basis Construction Scheme

Set $B := \emptyset$. For each $1 \leq i \leq w$, distinguish the following two cases:

1. $B = \emptyset$: if $\mathbf{x}_i \in C$, then do nothing, otherwise, then add $\mathbf{x}_i$ in B.
2. Otherwise: let $B = \{\mathbf{x}_{g_1}, \ldots, \mathbf{x}_{g_b}\}$ $(b \geq 1)$ where $1 \leq g_1 < \ldots < g_b < i$, if there exist $a_1, \ldots, a_b \in \mathbb{F}$ such that $\mathbf{x}_i + a_1\mathbf{x}_{g_1} + \ldots + a_b\mathbf{x}_{g_b} \in C$, then do nothing; otherwise, add $\mathbf{x}_i$ in B.

Let B be the pseudo-basis. Thus $|B|$ is the pseudo-dimension. **End.**

Before we show how the decoder can decode the information of the codewords by analysing the pseudo-basis, we give the following lemma for further discussion.

Lemma 3.2.3. *For any codeword $\mathbf{c} = (c_1, \ldots, c_h) \in C$, let $T = \{i | c_i \neq 0\}$. If $D \notin 2\mathcal{A}$ and $\psi(T) \in \mathcal{A}$, then the information of $\mathbf{c}$ is 0.*

Proof. Let $O = \{i | c_i = 0\}$. From $D \notin 2\mathcal{A}$ and $\psi(T) \in \mathcal{A}$, we have $\psi(O) \in \Gamma$. Due to Lemma 3.2.1, the information of $\mathbf{c}$ can be decoded with all the entries c_i where $i \in O$. Since all these entries are 0's, the information of $\mathbf{c}$ is 0. □

Based on this Lemma, we describe a new concept called *invalid error vector*, which is crucial for the correctness of our decoding scheme.

Definition 3.2.6. *An* invalid error vector *is an error vector that is a codeword.*

Given a codeword $\mathbf{c} \in C$ and a vector $\mathbf{x}$, we let $\mathbf{e} = \mathbf{x} - \mathbf{c}$ be an error vector such that $\psi(E) \in \mathcal{A}$. If $\mathbf{e}$ is a codeword (i.e., $\mathbf{e} \in C$), then $\mathbf{x} \in C$. Due to Lemma 3.2.3, the information of the codeword $\mathbf{e}$ is 0, so the information of $\mathbf{x}$ equals to the information of $\mathbf{c}$. That is, the error vector $\mathbf{e}$ does not actually cause errors. Thus we call this kind of error vector *invalid*. Evidently, the vector $\mathbf{0} \in \mathbb{F}^h$ is an invalid error vector.[4]

Let $B = \{\mathbf{x}_{g_1}, \ldots, \mathbf{x}_{g_b}\}$ be a pseudo-basis and $E_{g_1}, \ldots, E_{g_b}$ be the respective error locators $(1 \leq g_1 < \ldots < g_b \leq w)$, we denote $F = \bigcup_{i=1}^{b} E_{g_i}$ as the *final error locator* of B.

Due to the existence of invalid error vectors, the final error locator may not indicate all the error locations. Indeed, it is possible that there exists a vector $\mathbf{x}_i \notin B$ such that its error locator $E_i \nsubseteq F$, so some locations in E_i is not identified by F. Next, we show that even though the final error locator may not indicate all the error locations, the decoder can still decode all the information reliably with it.

Lemma 3.2.4. *If the final error locator of a pseudo-basis is known, then the decoder can decode the information of all the codewords.*

Proof. Given the final error locator F of a pseudo-basis $B = \{\mathbf{x}_{g_1}, \ldots, \mathbf{x}_{g_b}\}$, a decoding scheme is as simple as follows:

Decoding Scheme from the pseudo-basis

For each $1 \leq i \leq w$, decode the information r_i of $\mathbf{c}_i$ from $\mathbf{x}_i$ such that if $j \in F$, then the j-th entry of $\mathbf{x}_i$ is not used for decoding. **End.**

It is straightforward that if $i \in \{g_1, \ldots, g_b\}$, then the decoded information r_i is correct. Indeed, $D \notin 2\mathcal{A}$ and $\psi(F) \in \mathcal{A}$ imply that $\psi(\{1, \ldots, h\} \setminus F) \in \Gamma$. Thus according to Lemma 3.2.1, the entries not indicated by F can be used to decode r_i. Since F contains all the error locations of $\mathbf{x}_i$, all the entries that are used to decode r_i are correct.

Next, if $i \in \{1, \ldots, w\} \setminus \{g_1, \ldots, g_b\}$, then as we discussed before this lemma, it is possible that $E_i \nsubseteq F$. That is, errors may exist in the entries used to decode r_i. Since $\mathbf{x}_i \notin B$, there exist $a_1, \ldots, a_b \in \mathbb{F}$ such that $\mathbf{x}_i + a_1\mathbf{x}_{g_1} + \ldots + a_b\mathbf{x}_{g_b} \in C$. Thus $\mathbf{e}_i + a_1\mathbf{e}_{g_1} + \ldots + a_b\mathbf{e}_{g_b} \in C$. Let $\mathbf{e}'_i = \mathbf{e}_i + a_1\mathbf{e}_{g_1} + \ldots + a_b\mathbf{e}_{g_b}$, we regard $\mathbf{e}'_i$ as an *invalid error vector*. Thus one can decode the information r_i of $\mathbf{c}_i$ correctly from the vector $\mathbf{x}'_i = \mathbf{c}_i + \mathbf{e}'_i$ because $\mathbf{x}'_i \in C$. Since $\mathbf{x}_i = \mathbf{c}_i + \mathbf{e}_i = \mathbf{x}'_i - (a_1\mathbf{e}_{g_1} + \ldots + a_b\mathbf{e}_{g_b})$, it is clear that excluding the entries indicated by F, the remaining entries of $\mathbf{x}_i$ are the same as those of $\mathbf{x}'_i$. That is, even though errors may exist in these remaining entries, one can decode the information r_i of $\mathbf{c}_i$ correctly from these entries. □

[4] Note that in the threshold model, where any (up to) t entries of the codeword of size $n \geq 2t+1$ can be errors, invalid error vectors (except $\mathbf{0} \in \mathbb{F}^n$) do not exist. This is because if $n = 2t+1$, then the minimum Hamming distance is $t+1$. Since $\mathbf{0} \in \mathbb{F}^n$ is always a codeword, the other codewords must have at least $t+1$ non-zero entries. Thus any error vector with at most t non-zero entries is not a codeword.

It is trivial that the pseudo-dimension $|B|$ of our scheme is at most $wt^{\mathcal{A}} = O(h)$. This is because the dimension of the vector space spanned by the error vectors is $wt^{\mathcal{A}}$, since there are at most $wt^{\mathcal{A}}$ non-zero entries in each error vector. Therefore, the pseudo-basis has $O(h^2)$ field elements.

Using the generalized linear code and the idea of pseudo-basis and pseudo-dimension, it is possible to design efficient PSMT ($(0,0)$-SMT) protocols with constant round complexity (RC). We show our results later in Chapter 4.

3.3 Critical Paths

Unlike those in the threshold model, the conditions for SMT in the general adversary model do not require node-disjoint paths. This raises the question of how the messages are transmitted in a general network graph. A straightforward solution (though somehow less efficient) is to characterize the graph into all possible paths between S and R, and the messages are transmitted through these paths. To this end, the idea of *critical paths* was introduced by Kumar et al. [KGSR02] in their initial study (see Appendix A.1). We extend their study, by first giving a formal definition as follows.

Definition 3.3.1. *Given a graph $G(V, E)$, in which S and R are connected with a certain connectivity, say ξ-connectivity. A set of paths P is called* critical *if S and R remain ξ-connected with all paths in P, but lose the ξ-connectivity with all paths in any set $P' \subsetneq P$. Let $\mathcal{H}$ be the set of all critical sets of paths, we define a* minimal critical set *P^* as $P^* \in \mathcal{H}$ such that $|P^*| = \min\{|P| : P \in \mathcal{H}\}$.*

For example, if S and R are $2\mathcal{A}$-connected in a graph $G(V, E)$ (see Definition 2.2.6), where $\mathcal{A}$ is an adversary structure, then a set of paths P is called critical if, S and R remain $2\mathcal{A}$-connected with all paths in P, but they are $2\mathcal{A}$-separated with the paths in any set $P' \subsetneq P$. Our definition of critical paths is general, as it can be applied to any kind of connectivity.

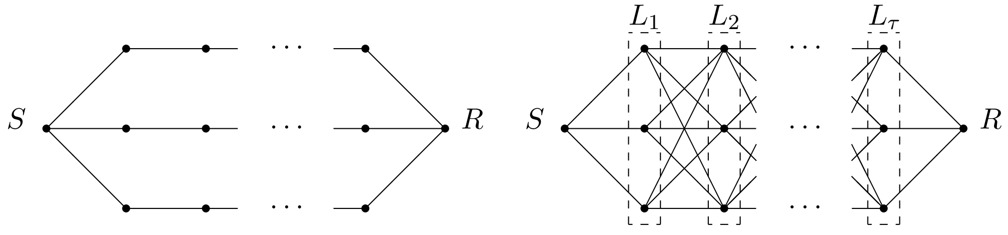
Without loss of generality, we assume that there does not exist a trusted path between S and R; i.e., $|P^*| > 1$. We show the following observation.

Observation 3.3.1. *With any graph in which S and R are $d\mathcal{A}$-connected, $|P^*|$ can be as small as $d + 1$ or as large as exponential in the size of the graph.*

We show two examples in Figure 3.1. In the examples we assume that S and R are $2\mathcal{A}$-connected, where $\mathcal{A}$ is an adversary structure on $V \setminus \{S, R\}$.

First, consider a graph G_1 as shown in Figure 3.1(a), in which there are only three paths between S and R. The adversary structure $\mathcal{A}$ has the following property: all nodes in any set $A \in \mathcal{A}$ are on the same path. Thus it is clear that in G_1, S and R are $2\mathcal{A}$-connected, and all the three paths are in P^*.

Next, consider a graph G_2 as shown in Figure 3.1(b). We assume that except S and R, there are 3τ nodes in G_2. We can view S and R as they are connected by τ



(a) G_1: $|P^*| = d+1 = 3$. **(b)** G_2: $|P^*|$ is exponential in n.

Figure 3.1: $2\mathcal{A}$-connectivity in different graphs.

levels $L_1, \ldots, L_\tau$, where each level L_i $(1 \le i \le \tau)$ is a set of three nodes, and there is an edge between each node in L_i and each node in L_{i+1}. The adversary structure $\mathcal{A}$ has the following property: for each set $A \in 2^{V \setminus \{S,R\}}$, if there exist two nodes $v_1, v_2 \in A$ such that $v_1, v_2 \in L_i$ $(1 \le i \le \tau)$, then $A \notin \mathcal{A}$; otherwise $A \in \mathcal{A}$. In other words, the adversary can control at most 1 node of each level. Obviously, S and R are $2\mathcal{A}$-connected in G_2, but if we remove any edge from the graph, then they are $2\mathcal{A}$-separated. Also straightforwardly $|P^*| = 3^\tau$, because the critical paths are all the paths on each of which there is exactly one node of each level, and there are 3^τ such paths. Thus we have that $|P^*|$ is exponential in the size of the graph, which is $9\tau - 3$.

To sum up, the two examples in Figure 3.1 are used to demonstrate Observation 3.3.1. Evidently, our examples can easily be adapted to other connectivities; e.g., the $3\mathcal{A}$-connectivity.

Therefore, if an SMT protocol is executed via the paths in a graph, then it is impossible to determine its CC in the size of the graph, because the number of paths varies remarkably in different graphs with the same connectivity (e.g., G_1 and G_2). Thus in this thesis, *we determine CC in the number of critical paths.*

Now we define our model of critical paths to characterize the network graph.

Definition 3.3.2. *Given a graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A} = \{A_1, \ldots, A_z\}$ be an adversary structure on $V \setminus \{S, R\}$, a* Basic Characterization *of G given $\mathcal{A}$ is defined as follows:*

- *If G is an undirected graph, then we use P to denote a minimal critical set of paths between S and R, and we write $P_1, \ldots, P_z$, where $P_i \subseteq P$ for each $1 \le i \le z$, to denote the sets of the paths in P that $A_1, \ldots, A_z$ cut respectively.*

- *If G is a directed graph, then we use P and Q to denote the minimal critical sets of the forward and feedback paths respectively, and we write $P_1, \ldots, P_z$ ($Q_1, \ldots, Q_z$), where $P_i \subseteq P$ ($Q_i \subseteq Q$) for each $1 \le i \le z$, to denote the sets of the forward (feedback) paths in P (Q) that $A_1, \ldots, A_z$ cut respectively.*

Furthermore, in our study of SMT in point-to-point networks, to effectively use critical paths, we construct new structures which consist of the subsets of critical paths.

Definition 3.3.3. *Given a graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A} = \{A_1, \ldots, A_z\}$ be an adversary structure on $V \setminus \{S, R\}$.*

- *If G is an undirected graph, then $\mathcal{P} = \{P_1, \ldots, P_z\}$ is the* critical-path structure *on P.*
- *If G is a directed graph, then $\mathcal{P} = \{P_1, \ldots, P_z\}$ and $\mathcal{Q} = \{Q_1, \ldots, Q_z\}$ are the* critical-path structures *on P and Q respectively. We also denote the* critical-path structure *$\mathcal{P} \diamond \mathcal{Q} = \{P_1 \cup Q_1, \ldots, P_z \cup Q_z\}$.*

In the next chapter, we study in more depth the properties of the critical-path structures, and show how SMT protocols can be constructed with respect to the critical-path structures.

3.4 Brief Conclusion of Chapter 3

In this chapter, we prepared our study of SMT with some ideas and observations. In Section 3.1, we proved that an almost perfect threshold $(t+1, n, \kappa)$-SSS can correct the same number of errors as a perfect $(t+1, n, 0)$-SSS can. This result will be used to determine the necessary condition for almost perfect private communication. Next in Section 3.2, we presented a generalized linear code with error-correcting capability, and generalized the idea of pseudo-basis and pseudo-dimension, which will later be used for efficient PSMT protocol design. Finally in Section 3.3, we showed our observation regarding critical paths in a general network graph, and defined the Basic Characterization of the graphs and new adversary structures over critical paths.

In the next chapter, we study SMT in point-to-point networks. All the results obtained in this chapter will then be used.

Chapter 4

SMT in Point-to-Point Networks

In this chapter we address the problem of secure message transmission (SMT) in point-to-point networks. As we discussed in Section 2.2.1, different point-to-point networks can be modelled by either undirected or directed graphs. Here we study both. The work in this chapter is devoted to determine minimal network connectivities for different levels of almost perfectly secure message transmission (APSMT), and also to design efficient perfectly secure message transmission (PSMT) protocols. Besides this, the first result in this chapter is a cryptanalysis of some previous PSMT protocols.

We first focus on PSMT in directed graphs with feedback paths from R to S. The study of this setting was initiated by Desmedt and Wang in [DW02] and then followed by Patra et al. in [PSC$^+$07]. We observe that all their "PSMT protocols" do not guarantee perfect privacy, when the adversary performs a delicately designed *Guessing Attack*. In Section 4.1, we describe such a Guessing Attack against these existing protocols.

Next, we determine minimal network connectivities for APSMT in both undirected and directed graphs. We study δ-RMT and $(0, \delta)$-SMT in the general adversary model, and also show that reducing the level of privacy does not weaken the connectivity requirements. Our results should complete Table 2.1, and end our search for minimal connectivities for SMT in the point-to-point setting. These results will be presented in Section 4.2.

Finally, we study PSMT in both undirected and directed graphs. We show that using the generalized linear code we discussed in Section 3.2, efficient PSMT protocols can be designed to significantly reduce the communication costs of the previous protocols (e.g., in [KGSR02, YD09]) in the general adversary model. We present these results in three sections: Section 4.3 gives the preliminaries for PSMT protocols and shows how our protocols would improve the previous results; respectively in Section 4.4 and Section 4.5, we give efficient protocols for PSMT in undirected and directed graphs.

Some results in this chapter has been published in [YD09, YD10].

4.1 Guessing Attack

In this section we study a particular point-to-point network setting: a directed graph with feedback paths. In general, the feedback paths are used by the receiver R to seek for help from the sender S when R does not have enough information to recover the messages (i.e., for reliability purposes). However, this emphasis on reliability may lead to a damage on perfect privacy. Indeed, we observe that all PSMT (i.e., (0,0)-SMT) protocols proposed by Desmedt and Wang in [DW02] and Patra et al. in [PSC$^+$07] do not guarantee 0-privacy. To breach 0-privacy of these protocols, we design a *Guessing Attack* which can be performed by the adversary on the feedback paths.

The basic idea of the Guessing Attack is to replace the feedbacks from R to S on the feedback paths with something that may reveal the message. This guess will be successful with some probability. Even if the guess is not correct, the probability of learning the actual secret is better than random guessing, so perfect privacy of the protocol is violated.

In the following, we show how our Guessing Attack can be designed as the strategy of the adversary against some existing protocols.

4.1.1 On Desmedt and Wang's Protocols

Here we give an example of how the Guessing Attack breaches 0-privacy of one of Desmedt and Wang's protocols in [DW02]. This DW protocol (the protocol of [DW02, Theorem 5]) is for $(0,0)$-SMT against a t-bounded adversary in the threshold model. First we sketch the DW protocol as follows.

Condition for the DW protocol: There are $3t \geq 2t+1$ directed node-disjoint paths from S to R and one directed node-disjoint path from R to S.[1]

Sketch of the DW protocol Let $p_1, ..., p_{3t}$ be the directed paths from S to R and q be the directed path from R to S.

Step 1 ...

Step 2 S chooses a $key^S \in_R \mathbb{F}$ and uses a $(t+1, 3t, 0)$-SSS to construct shares $\mathbf{c} = (s_1, ..., s_{3t})$ of key^S. For each $1 \leq i \leq 3t$, S sends s_i to R via path p_i.

Step 3 Let $\mathbf{c}^R = (s_1^R, ..., s_{3t}^R)$ be the shares R receives. If R finds that there are at most $t-1$ errors (using error-correcting code: see Section 2.7), then R recovers key^R from the shares, and sends 'stop' to S via path q; otherwise, R sends $\mathbf{c}^R$ to S via path q.

Step 4 If S receives $\mathbf{c}^S = (s_1^S, ..., s_{3t}^S)$ from path q, S broadcasts $F = \{i : s_i^S \neq s_i\}$ ($|F| = t$) via all paths $p_1, ..., p_{3t}$; otherwise, S broadcasts 'stop'.

[1] This condition is sufficient for $(0,0)$-secure message transmission from S to R, but is stronger than the necessary condition, which is $n \geq \max\{3t+1-2u, 2t+1\}$ [DW02], where n is the number of the forward paths and u is the number of the feedback paths. See Table 2.1 for more details.

Step 5 . . .

Step 6 S broadcasts $m + key^S$ via all paths $p_1, ..., p_{3t}$, where m is the actual message.

Step 7 . . .

This single-feedback-path protocol is the basis of the main protocols in [DW02]. We observe that this DW protocol is 0-reliable, so in the above sketch we did not describe how R recovers the message (see [DW02] for the complete protocol). Now we show that using our Guessing Attack, the adversary can learn the message m with some extra information it gets during the execution of the protocol.

Theorem 4.1.1. *This DW protocol is not a* 0*-private message transmission protocol from S to R.*

Proof. Let X be a variable induced by key^S and adv be the view of the adversary through this DW protocol, due to the fact that $key^S \in_R \mathbb{F}$, if the protocol is 0-private, then the probability that the adversary learns key^S is $\Pr[X = key^S|adv] = \Pr[X = key^S] = \frac{1}{|\mathbb{F}|}$. Now we show a Guessing Attack by which the adversary can learn key^S with probability better than $\frac{1}{|\mathbb{F}|}$.

Guessing Attack on the DW protocol

> The t-bounded adversary chooses to control forward paths $p_1, \dots, p_{t-1}$ and feedback path q. Thus the adversary is able to get shares $s_1, \dots, s_{t-1}$ in Step 2. With these $t - 1$ shares, the adversary performs as follows.
>
> The adversary chooses a share $s'_t \in_R \mathbb{F}$ and two keys $key'_1, key'_2 \in_R \mathbb{F}$ where $key'_1 \neq key'_2$. Regarding key'_1, the adversary assumes that $s_1, \dots, s_{t-1}, s'_t$ are t shares of key'_1, thus using Lagrange interpolation, the adversary outputs the other t shares $s'_{t+1}, \dots, s'_{2t}$ of key'_1. Similarly, regarding key'_2, the adversary assumes that $s_1, \dots, s_{t-1}, s'_t$ are t shares of key'_2, and outputs the other t shares $s'_{2t+1}, \dots, s'_{3t}$ of key'_2. The adversary sets $\mathbf{c}' = (s_1, \dots, s_{t-1}, s'_t, \dots, s'_{3t})$.
>
> In each execution step of the DW protocol, the adversary acts in a passive manner on the forward paths $p_1, \dots, p_{t-1}$. Thus R sends 'stop' to S in Step 3. On the feedback path q, the adversary ignores what R sends and transmits $\mathbf{c}'$ to S. Then in Step 4, if S finds exactly t errors in $\mathbf{c}^S$, which is actually $\mathbf{c}'$, then S broadcasts $F = \{i : s'_i \neq s_i\}$, according to which the adversary can recover $key^S = key'_j$ $(j \in \{1, 2\})$; otherwise, S broadcasts 'stop', and the adversary chooses a $key' \in_R \mathbb{F}$, as its key^S. **End.**

In this Guessing Attack, the adversary guesses a share s'_t and two keys key'_1 and key'_2. Now S will broadcast the set F if S finds exactly t errors in $\mathbf{c}'$, and the t errors can only be either $s'_{t+1}, \dots, s'_{2t}$ or $s'_{2t+1}, \dots, s'_{3t}$. That is, the guess is successful if $s'_t = s_t$

and one of the two keys is correct; i.e., $key'_j = key^S$ $(j \in \{1, 2\})$. Thus the probability η that the guess in successful is

$$\eta = \frac{1}{|\mathbb{F}|} \times \left(2 \times \frac{1}{|\mathbb{F}|}\right) = \frac{2}{|\mathbb{F}|^2} \ .$$

If the guess fails, then the adversary will choose a $key' \in_R \mathbb{F}$, and with probability $\frac{1}{|\mathbb{F}|}$, $key' = key^S$. Thus, the total probability θ that the adversary learns key^S by performing the Guessing Attack is

$$\theta = \eta + (1 - \eta) \times \frac{1}{|\mathbb{F}|} > \frac{1}{|\mathbb{F}|} \ .$$

Therefore, the adversary can learn key^S with a probability better than $\frac{1}{|\mathbb{F}|}$, and hence can recover the message m with better probability.[2] Thus this DW protocol is not 0-private. □

This example clearly shows how a Guessing Attack can be performed on the feedback paths. In the journal paper [WD08], Wang and Desmedt provided a fixed protocol that uses induction when S receives doubtable feedbacks (i.e., the case that a Guessing Attack may happen). The new protocol is 0-private. For more details of the (0,0)-SMT protocol tolerating a threshold adversary, we refer to [WD08, Theorem 4.2].

Next we show our Guessing Attack on a previous protocol for PSMT in the general adversary model, which is proposed by Patra et al. in [PSC$^+$07].

4.1.2 On Patra et al.'s Protocols

In [PSC$^+$07], Patra et al. proposed three protocols for (0,0)-SMT with feedbacks. Two of the protocols are in the threshold model, and the other one is in the general adversary model. We observe that neither of these three protocols achieves 0-privacy when the Guessing Attack takes place. Here we show a Guessing Attack on their so-called *Secure Protocol*, which was claimed to be (0,0)-secure tolerating a general adversary structure. For the attack on the other two protocols in the threshold model, we refer to [YD09].

First, we sketch the Secure Protocol. The Secure Protocol is a 3-round protocol tolerating a set $\mathcal{B}$, which is a subset of an adversary structure $\mathcal{A}$, where $|\mathcal{B}| = 3$. The Secure Protocol is actually a sub-protocol for (0,0)-SMT, and the whole protocol can be resembled with quasi-polynomial (in $|\mathcal{A}|$) number of instances of the Secure Protocol, using the sub-protocol reconstruction scheme described in Appendix A.5. Here we show that this sub-protocol itself does not achieve perfect privacy.

Conditions for Secure Protocol Let $\mathcal{B} = \{A_1, A_2, A_3\}$, S and R are strongly $3\mathcal{B}$-directed-connected (see Definition 2.2.8). It is straightforward that if S and R are $3\mathcal{B}$-connected with the forward paths, then $(0, 0)$-SMT is easy to achieve

[2] Although the message m may be chosen with respect to any probability distribution (not necessarily uniform), more knowledge on the key key^S gives better probability to recover m.

(see [DWB05]). In this protocol, we assume that $A_1 \cup A_2 \cup A_3$ cuts all the forward paths.

Sketch of Secure Protocol Since S and R are strongly $3\mathcal{B}$-directed-connected, they are $2\mathcal{B}$-connected on the forward paths, thus there exist 3 forward paths p_1, p_2, p_3 such that $p_1 \in P \setminus (P_2 \cup P_3)$, $p_2 \in P \setminus (P_1 \cup P_3)$ and $p_3 \in P \setminus (P_1 \cup P_2)$. Also since $A_1 \cup A_2 \cup A_3$ cuts all the forward paths, due to the strong $3\mathcal{B}$-directed-connectivity, we know that at most one of A_1, A_2, A_3 cuts all the feedback paths, thus there exist 2 feedback paths $q_\alpha \in Q \setminus Q_\alpha$ and $q_\beta \in Q \setminus Q_\beta$, where $\alpha, \beta \in \{1, 2, 3\}$.[3] Let m be the message S transmits to R:

Round 1 S chooses a bivariate polynomial $f(x, y) = \sum_{i=0}^{1} \sum_{j=0}^{1} r_{i,j} x^i y^j$ uniformly at random such that $f(0, 0) = m$. $f(x, y)$ is also *symmetric*; i.e., $f(i, j) = f(j, i)$. S sends the polynomial $f(x, i)$ to R via path p_i, $1 \leq i \leq 3$.

Round 2 R receives the polynomial $f_i^R(x) = f^R(x, i)$ on path p_i, $1 \leq i \leq 3$. Out of the three $f_i^R(x)$'s, at most one is corrupted. R then performs tests to determine which path p_i is faulty.[4] According to the outcome of the tests:

- if R concludes that all p_i's ($1 \leq i \leq 3$) are honest, then R recovers m and terminates the protocol;
- if R finds which p_i ($1 \leq i \leq 3$) is faulty, then R recovers m and terminates the protocol;
- if R finds that one of the two paths p_i and p_j ($1 \leq i, j \leq 3$ and $i \neq j$) is faulty but cannot distinguish which one, then R sends a 4-vector $(i, j, f_i^R(j), f_j^R(i))$ to S via paths q_α and q_β.

Round 3 S receives two 4-vectors: $(i_\alpha, j_\alpha, v_{i_\alpha}, v_{j_\alpha})$ on path q_α and $(i_\beta, j_\beta, v_{i_\beta}, v_{j_\beta})$ on path q_β.

- Regarding $(i_\alpha, j_\alpha, v_{i_\alpha}, v_{j_\alpha})$, S checks whether $v_{i_\alpha} = f(j_\alpha, i_\alpha)$ and whether $v_{j_\alpha} = f(i_\alpha, j_\alpha)$. According to the outcome, S identifies which path p_{i_α} or p_{j_α} is faulty, and appends an error message "Path γ is faulty" (γ is either p_{i_α} or p_{j_α}) to $(i_\alpha, j_\alpha, v_{i_\alpha}, v_{j_\alpha})$.
- S performs similar computation to the other 4-vector $(i_\beta, j_\beta, v_{i_\beta}, v_{j_\beta})$.
- S broadcasts the two 4-vectors along with the appended error messages via all paths in P.

... (We do not need to show how the message is recovered here. For the complete protocol, see [PSC+07].)

Next, we show that by performing a Guessing Attack, the adversary can breach perfect privacy of the Secure Protocol.

[3] See related notations of P, Q in Definition 3.3.2.

[4] The details of the tests are not important here, instead we refer to [PSC+07, Secure Protocol].

Theorem 4.1.2. *The Secure Protocol is not a* 0*-private message transmission protocol from S to R.*

Proof. Without loss of generality, we assume that $m \in_R \mathbb{F}$. Let X be the variable induced by m and adv be the view of the adversary through the Secure Protocol. If the protocol is 0-private, then the probability that the adversary learns m is $\Pr[X = m|adv] = \Pr[X = m] = \frac{1}{|\mathbb{F}|}$. Assuming that there exist two feedback paths $q_1 \in Q \setminus Q_1$ and $q_2 \in Q \setminus Q_2$ (i.e., $\alpha = 1$ and $\beta = 2$), and $q_1, q_2 \in Q_3$, we show a Guessing Attack as follows.

Guessing Attack on the Secure Protocol

> The adversary chooses Z_3 to control, so it corrupts both q_1 and q_2. In Round 1, the adversary can only get $f(x, 3)$, with which it knows $f(1, 3)$ and $f(2, 3)$. Because $f(x, y)$ is symmetric, we have $r_{0,1} = r_{1,0}$, thus using polynomial interpolation, we know that the adversary only needs $f(1, 2)$ to recover the message m. In each round of the Secure Protocol, the adversary acts in a passive manner on paths in P_3. Thus R does not use the feedback channel throughout the protocol. In Round 2, the adversary chooses 4 distinct random numbers $v'_1, v'_2, v'_3, v'_4 \in_R \mathbb{F}$, and transmits two 4-vectors $(1, 2, v'_1, v'_2)$ and $(1, 2, v'_3, v'_4)$ to S. Then in Round 3, if regarding a value v'_i $(1 \leq i \leq 4)$, no appended error message "Path γ is faulty" (γ is either p_1 or p_2) is broadcast by S, then the adversary knows that v'_i is correct (i.e., $v'_i = f(1, 2)$), and hence recovers m; otherwise, the adversary chooses $v' \in_R \mathbb{F} \setminus \{v'_1, v'_2, v'_3, v'_4\}$ as $f(1, 2)$ to recover a message m'. **End.**

In this Guessing Attack, the guess is successful if there is a $v'_i = f(1, 2)$ $(1 \leq i \leq 4)$, so the error messages S broadcasts in Round 3 will indicate which v'_i is correct. Thus the probability η that the guess is successful is

$$\eta = 4 \times \frac{1}{|\mathbb{F}|} = \frac{4}{|\mathbb{F}|} .$$

If the guess fails, then the adversary knows that neither of the 4 random numbers v'_1, v'_2, v'_3, v'_4 is correct. The adversary will choose a $v' \in_R \mathbb{F} \setminus \{v'_1, v'_2, v'_3, v'_4\}$, and with probability $\frac{1}{|\mathbb{F}|-4}$, $v' = f(1, 2)$, and the recovered message m' is the actual message; i.e., $m' = m$. Thus, the total probability θ that the adversary learns m by performing the Guessing Attack is

$$\theta = \eta + (1 - \eta) \times \frac{1}{|\mathbb{F}| - 4} = \frac{4}{|\mathbb{F}|} + \left(1 - \frac{4}{|\mathbb{F}|}\right) \times \frac{1}{|\mathbb{F}| - 4} = \frac{5}{|\mathbb{F}|} .$$

Therefore, the probability that the adversary learns m is much higher than expected (i.e., $\frac{1}{|\mathbb{F}|}$). Thus the Secure Protocol is not 0-private. □

We have just shown a Guessing Attack on an existing PRMT protocol in the general adversary model. To defend against such an attack, we propose a fixed protocol in a directed graph with feedback paths. This result will be presented in Section 4.5.

4.2 Minimal Connectivities for APSMT

In this section, we determine the necessary and sufficient conditions for different levels of almost perfect security. The results in this section consist of two parts: one on almost perfect reliability (see Section 4.2.1) and the other on almost perfect privacy (see Section 4.2.2). These results should finally complete Table 2.1.

4.2.1 Almost Perfect Reliability

Here we study almost perfect reliability to determine the necessary and sufficient conditions for δ-RMT and $(0, \delta)$-SMT in the general adversary model.

Against a t-bounded adversary in an undirected point-to-point graph, Franklin and Wright showed that given $\delta < \frac{1}{2}(1 - \frac{1}{|\mathbb{M}|})$, where $\mathbb{M} \subseteq \mathbb{F}$ is a message space, δ-RMT is possible if and only if $n \geq 2t + 1$ [FW00, Theorem 5.1]. Combining their proof with the technique of Desmedt et al. in [DWB05, Theorem 3], it is easy to prove that the following theorem is necessary.

Theorem 4.2.1. *Given an undirected graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, δ-RMT or $(0, \delta)$-SMT is possible if and only if S and R are $2\mathcal{A}$-connected in G.*

Since the $2\mathcal{A}$-connectivity is sufficient for 0-RMT and (0,0)-SMT, it is obviously sufficient for δ-RMT and $(0, \delta)$-SMT where the requirements for reliability are relaxed. Because lowering the requirement for reliability (by a negligible amount) does not weaken the connectivity in undirected graphs, there is no need to present protocols to show the sufficiency of the condition. We refer some related protocols to [KS09b, DESN10].

Next, we study δ-RMT and $(0, \delta)$-SMT in a directed graph. We use the same network model as that of [DW02], that is, the condition is expressed using the forward and feedback paths, and any heterogeneous paths are not considered. For the results on heterogeneous paths, we refer to [SR06, NAS11]. Now we consider the strong $2\mathcal{A}$-directed-connectivity described in Definition 2.2.9. Using the notation of Definition 3.3.3, we have that if S and R are strongly $2\mathcal{A}$-directed-connected, then $P \notin \mathcal{P}$ and $P \cup Q \notin 2(\mathcal{P} \diamond \mathcal{Q})$. Next we show that the strong $2\mathcal{A}$-directed-connectivity is the minimal connectivity for δ-RMT and $(0, \delta)$-SMT in directed graphs.

Theorem 4.2.2. *Given a directed graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, δ-RMT or $(0, \delta)$-SMT is possible if and only if S and R are strongly $2\mathcal{A}$-directed-connected in G.*

Proof. First, we show that the condition is necessary. It is straightforward that S and R must be $\mathcal{A}$-connected on the forward paths (i.e., $P \notin \mathcal{P}$), because there must exist one uncorrupted path from S to R. Moreover, $2\mathcal{A}$-connectivity with all the forward and feedback paths (i.e., $P \cup Q \notin 2(\mathcal{P} \diamond \mathcal{Q})$) is also necessary, because even if we strengthen the connectivity, such that all the forward and feedback paths are bi-directed, due to Theorem 4.2.1, δ-RMT or $(0, \delta)$-SMT is impossible if S and R are not $2\mathcal{A}$-connected. Therefore, the strong $2\mathcal{A}$-directed-connectivity is the lower bound.

Next, we show that the condition is sufficient for $(0, \delta)$-SMT, and hence it is sufficient for δ-RMT. Let $\mathcal{A} = \{A_1, \ldots, A_z\}$ and $m \in \mathbb{M}$ be the message, we construct a 2-round $(0, \delta)$-SMT protocol, which uses the authentication code we discussed in Section 2.8, as follows.

3-Round Directed APSMT Protocol

Round 1 - S to R and R to S in parallel: For each $1 \leq i \leq z$:

1. For each path $p_j \in P \setminus P_i$, S chooses $(a^S_{i,j}, b^S_{i,j}, c^S_{i,j}) \in_R \mathbb{F}^3$ and sends the 3-vector to R via path p_j.
2. For each path $q_j \in Q \setminus Q_i$, R chooses $(d^R_{i,j}, e^R_{i,j}, f^R_{i,j}) \in_R \mathbb{F}^3$ and sends the 3-vector to S via path q_j.
3. On each path $p_j \in P \setminus P_i$, R receives a 3-vector $(a^R_{i,j}, b^R_{i,j}, c^R_{i,j})$. On each path $q_j \in Q \setminus Q_i$, S receives a 3-vector $(d^S_{i,j}, e^S_{i,j}, f^S_{i,j})$.

Round 2 - S to R: For each $1 \leq i \leq z$:

1. S computes
$\alpha^S_i = \sum_{p_j \in P \setminus P_i} a^S_{i,j} + \sum_{q_j \in Q \setminus Q_i} d^S_{i,j}$,
$\beta^S_i = \sum_{p_j \in P \setminus P_i} b^S_{i,j} + \sum_{q_j \in Q \setminus Q_i} e^S_{i,j}$ and
$\gamma^S_i = \sum_{p_j \in P \setminus P_i} c^S_{i,j} + \sum_{q_j \in Q \setminus Q_i} f^S_{i,j}$.
2. S sends the pair $(m + \alpha^S, \mathrm{auth}(m + \alpha^S_i; \beta^S_i, \gamma^S_i))$ to R via all paths in $P \setminus P_i$.

Recovery Phase For each $1 \leq i \leq z$:

1. On each path $p_j \in P \setminus P_i$, R receives a pair $(g^R_{i,j}, h^R_{i,j})$.
2. If the pairs received on all paths in $P \setminus P_i$ are the same, we call this pair (g^R_i, h^R_i) then R computes
$\alpha^R_i = \sum_{p_j \in P \setminus P_i} a^R_{i,j} + \sum_{q_j \in Q \setminus Q_i} d^R_{i,j}$,
$\beta^R_i = \sum_{p_j \in P \setminus P_i} b^R_{i,j} + \sum_{q_j \in Q \setminus Q_i} e^R_{i,j}$ and
$\gamma^R_i = \sum_{p_j \in P \setminus P_i} c^R_{i,j} + \sum_{q_j \in Q \setminus Q_i} f^R_{i,j}$.
If $h^R_i = \mathrm{auth}(g^R_i; \beta^R_i, \gamma^R_i)$, then R recovers the message $m' = g^R_i - \alpha^R_i$, and terminates the protocol. **End.**

Now we show that the 3-Round Directed APMST Protocol is $(0, \delta)$-secure. Assume that the adversary chooses a set $A_e \in \mathcal{A}$ to control. We prove that the protocol is 0-private. For each $1 \leq i \leq z$, the keys are transmitted via the paths in $(P \cup Q) \setminus (P_i \cup Q_i)$.

Since S and R are $2\mathcal{A}$-connected with all paths in $P \cup Q$, there always exists a (forward or feedback) path $w_j \in (P \cup Q) \setminus ((P_i \cup Q_i) \cup (P_e \cup Q_e))$. Thus the adversary will not learn the keys transmitted on w_j, and hence it cannot compute $\alpha_i^S, \beta_i^S, \gamma_i^S$ to recover m. Thus 0-privacy is guaranteed.

Finally, we prove that the protocol is δ-reliable. It is straightforward that if $i = e$ (i.e., the adversary chooses $A_i \in \mathcal{A}$ to control), then the message $m' = m$ can be reliably recovered. This is because all the keys used for authentication, which are transmitted via $(P \cup Q) \setminus (P_i \cup Q_i)$, are uncorrupted. Next, for any $1 \leq i \leq z$, we use RT to denote the event that the message m' recovered by R is correct (i.e., $m' = m$), and use $\overline{RT}$ to denote the event otherwise. In order for m' to be recovered, the pairs received on all paths in $P \setminus P_i$ must be the same, thus in the event $\overline{RT}$, we have $P_i \cup P_e = P$. Similar to what we showed above, we know that the adversary does not have enough information to compute $\alpha_i^R, \beta_i^R, \gamma_i^R$. Thus $\overline{RT}$ only happens if the adversary modifies g_i^R so that $g_i^R \neq m + \alpha_i^S$ and successfully guesses an element $h_i^R \in_R \mathbb{F}$ such that $h_i^R = \text{auth}(g_i^R; \beta_i^R, \gamma_i^R)$. Thus $\Pr[\overline{RT}] = \frac{1}{|\mathbb{F}|}$ is the probability that the guess of h_i^R is successful. Since $1 \leq i \leq z$, the adversary has at most z attempts to make its guesses successful. That is, the probability that the adversary can breach the reliability of the protocol with successful guesses is

$$\delta = z \times \Pr[\overline{RT}] = z \times \frac{1}{|\mathbb{F}|} = \frac{z}{|\mathbb{F}|}.$$

It is straightforward that δ can be made negligible in the security parameters (given $\mathbb{F}$ is sufficiently large). Hence we have the proof. □

Therefore, we showed the necessary and sufficient conditions for almost perfect reliability when no privacy or perfect privacy is guaranteed. In the next section, we study almost perfect privacy.

4.2.2 Almost Perfect Privacy

The main result in this section shows that reducing the requirement for privacy does not weaken the minimal network connectivity. That is, the necessary and sufficient conditions for ϵ-privacy are the same as those for 0-privacy. This result is general, regardless of the adversary models and network settings. Proving the conditions for all the network and adversary models will be redundant. We therefore present only one example. That is, we prove that in a directed graph, the condition for (0,0)-SMT—the strong $3\mathcal{A}$-directed-connectivity (see Definition 2.2.8)—is also necessary and sufficient for $(\epsilon, 0)$-SMT. The technique used in this proof should be enough to show how similar results can be obtained in other network and adversary models.

Theorem 4.2.3. *Given a directed graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, $(\epsilon, 0)$-SMT is possible if and only if S and R are strongly $3\mathcal{A}$-directed-connected in G.*

Proof. The sufficiency of the condition is straightforward. As shown in the proof of Theorem 4.1.2, Patra et al.'s Secure Protocol in [PSC$^+$07] is actually an $(\epsilon, 0)$-SMT protocol. Next, using a technique similar to that in [DW02, DWB05], we prove the necessity of the condition.

According to Table 2.1, the $2\mathcal{A}$-connectivity on the forward paths is necessary for 0-RMT, and hence it is necessary for $(\epsilon, 0)$-SMT. Now we show that $(\epsilon, 0)$-SMT is possible only if for any three sets $A_1, A_2, A_3 \in \mathcal{A}$, if $A_1 \cup A_2 \cup A_3$ cuts all the forward paths, then at most one of these three sets cuts all the feedback paths.

For contradiction, we assume that there are three sets $A_1, A_2, A_3 \in \mathcal{A}$ such that $P_1 \cup P_2 \cup P_3 = P$, $Q_1 \subsetneq Q$ and $Q_2 = Q_3 = Q$, and there exists a $(\epsilon, 0)$-SMT protocol Π in this graph G. Let $m \in \mathbb{M}$ be the message that S wants to send to R. The adversary will simulate the possible behaviours of S and R by executing Π to transmit another message $m' \in \mathbb{M}$. The strategy of the adversary is to choose an $e \in_R \{1, 2, 3\}$ and control the set A_e. During the execution of the protocol Π, the adversary acts as follows:

- If $e = 1$, then the adversary acts in a passive manner throughout the protocol.
- If $e = 2$, then the adversary corrupts P_2. On all paths in P_2, the adversary ignores what S sends in each step of Π and simulates the protocol as S sent the message m'. On all paths in $Q_2 = Q$, the adversary ignores what R sends in each step of Π and simulates what R would send to S if $e = 1$.
- If $e = 3$, then the adversary corrupts $P \setminus (P_1 \cup P_2)$, which is possible because $P_1 \cup P_2 \cup P_3 = P \Rightarrow P \setminus (P_1 \cup P_2) \subseteq P_3$. On all paths in $P \setminus (P_1 \cup P_2)$, the adversary ignores what S sends in each step of Π and simulates the protocol as S sent the message m'. On all paths in $Q_3 = Q$, the adversary ignores what R sends in each step of Π and simulates what R would send to S if $e = 1$.

Note that the simulation of the adversary on the feedback paths in Q when $e = 2$ or $e = 3$ may not be successful, because R may send something that the adversary fails to learn. However, there is a non-zero probability with which the simulation succeeds, given the adversary knows the protocol and can always guess. In the following, we consider the case that this simulation succeeds.

It is straightforward that despite the value of e, the feedbacks that S receives are the same, thus the view of S is always the same. At the end of the protocol, the view of R can be divided into three parts $view_1$, $view_2$ and $view_3$, where $view_2$ consists of all the information the paths in P_2 have learned, $view_3$ consists of all the information the paths in $P \setminus (P_1 \cup P_2)$ have learned and $view_1$ consists of all the information the rest of the paths in P have learned. Given $e \in_R \{1, 2, 3\}$, let $adv(m, r)$ be the adversary's view when the actual message is m, since Π is ϵ-private, we have that for any two messages $m_0, m_1 \in \mathbb{M}$,

$$\textstyle\sum_{view_e} |\Pr[adv(m_0, r) = view_e] - \Pr[adv(m_1, r) = view_e]| \leq 2\epsilon < 1.$$

That is, let X be a variable induced by m, for any message $m_0 \in \mathbb{M}$ we have

$$\Pr[X = m_0 | view_e] \leq \Pr[X = m_0] + \kappa < 1.$$

Moreover, since Π is 0-reliable, R should be able to recover the actual message m from any two of the views $view_1, view_2, view_3$ with probability 1. Thus for any $\{i, j\} \subset \{1, 2, 3\}$, we have

$$\Pr[X = m | view_i, view_j] = 1.$$

Thus we regard $view_1, view_2, view_3$ as shares of m distributed using a $(2, 3, \kappa)$-SSS (see Definition 2.6.1). Now when $e = 2$ or $e = 3$, R should be able to distinguish which view of $view_2$ or $view_3$ is corrupted. To sum up, $view_1, view_2, view_3$ are shares of a $(2, 3, \kappa)$-SSS, and this SSS can correct 1 error: either $view_2$ or $view_3$. Due to Theorem 3.1.2, a $(2, 3, \kappa)$-SSS can only correct $\lfloor \frac{3-1-1}{2} \rfloor = 0$ error. We have a contradiction. □

Next, it is straightforward that the following result can be obtained using similar proofs to that of Theorem 4.2.3.

Corollary 4.2.1. *Let $0 \leq \delta < \frac{1}{2}$ and $0 \leq \epsilon_1 < \epsilon_2 < 1$, in any network model and any adversary model, the minimal connectivity required for (ϵ_1, δ)-SMT is the same as that for (ϵ_2, δ)-SMT.*

Therefore, in the following, we show the final results for ϵ-private message transmission in different network and adversary models.

Corollary 4.2.2. *Given an undirected graph $G(V, E)$ where $S, R \in V$, let n be the number of paths between S and R:*

- *In the threshold model where the adversary is bounded by t nodes of $V \setminus \{S, R\}$, (ϵ, δ)-SMT or $(\epsilon, 0)$-SMT is possible if and only if $n \geq 2t + 1$.*
- *In the general adversary model where the adversary is characterized by an adversary structure $\mathcal{A}$ on $V \setminus \{S, R\}$, (ϵ, δ)-SMT or $(\epsilon, 0)$-SMT is possible if and only if S and R are $2\mathcal{A}$-connected in G.*

Corollary 4.2.3. *Given a directed graph $G(V, E)$ where $S, R \in V$, let n be the number of forward paths and u be the number of feedback paths:*

- *In the threshold model where the adversary is bounded by t nodes of $V \setminus \{S, R\}$, (ϵ, δ)-SMT is possible if and only if $n \geq \max\{2t + 1 - u, t + 1\}$, and $(\epsilon, 0)$-SMT is possible if and only if $n \geq \max\{3t + 1 - 2u, 2t + 1\}$.*
- *In the general adversary model where the adversary is characterized by an adversary structure $\mathcal{A}$ on $V \setminus \{S, R\}$, (ϵ, δ)-SMT is possible if and only if S and R are strongly $2\mathcal{A}$-directed-connected in G, and $(\epsilon, 0)$-SMT is possible if and only if S and R are strongly $3\mathcal{A}$-directed-connected in G.*

Therefore, together with the results given in Section 4.2.1, we can complete Table 2.1 to give the minimal network connectivities for SMT in all kinds of point-to-point networks in different adversary models. We shall present a completed table in the concluding Chapter 6 at the end of this thesis.

4.3 PSMT Preliminaries

In the rest of the chapter, we study PSMT (i.e., (0,0)-SMT) protocols. Efficient PSMT protocols in undirected and directed graphs will be given in Section 4.4 and Section 4.5 respectively. In this section, we discuss some preliminaries so our protocols can be constructed in a clear manner.

First, we employ the model of critical paths defined in Definition 3.3.2 and the critical-path structure defined in Definition 3.3.3. That is:

- In an undirected graph, we denote $P = \{p_1, \ldots, p_n\}$ as a critical set of undirected paths, and denote $\mathcal{P} = \{P_1, \ldots, P_z\}$ as a critical-path structure on P. Thus n is the number of critical paths and $z = |\mathcal{P}| = |\mathcal{A}|$.
- In a directed graph, we denote $P = \{p_1, \ldots, p_n\}$ as a critical set of forward paths and $Q = \{q_1, \ldots, q_u\}$ as a critical set of feedback paths, and denote $\mathcal{P} = \{P_1, \ldots, P_z\}$ and $\mathcal{Q} = \{Q_1, \ldots, Q_z\}$ as critical-path structures on P and Q respectively. Thus n and u are the number of forward and feedback paths respectively and $z = |\mathcal{P}| = |\mathcal{Q}| = |\mathcal{A}|$. Without loss of generality, we assume $u = O(n)$.

Next, we show how our protocols would improve the previous results in terms of communication complexity (CC) and round complexity (RC) (see notations in Section 2.5).

4.3.1 Improvements to the Previous Results

Here we compare our results with the previous PSMT protocols in the general adversary model. As shown in Section 2.9.1, PSMT protocols tolerating adversary structures have been proposed by Kumar et al. [KGSR02], Desmedt et al. [DWB05], Patra et al. [PSC$^+$07] and Yang and Desmedt [YD09] for different network settings.[5] We show that our protocols in this work comprehensively improve the previous results in terms of communication complexity (CC) and round complexity (RC).

Because the previous protocols use different models for PSMT, it is not straightforward to compare their CC to our results. In fact, we need to compare the three parameters (n, z, h) that determine the CC of the protocols, where n is the number of critical paths, $z = |\mathcal{A}|$ and h is the size of the LSSS as well as the length of the codewords

[5] As discussed in Section 4.1, all protocols of Patra et al. in [PSC$^+$07] are vulnerable to the Guessing Attack. In [YD09], a fixed protocol is proposed against the Guessing Attack. However, this protocol still uses Patra et al.'s sub-protocol reconstruction scheme (see Appendix A.5). Thus as Patra et al.'s protocol, this fixed protocol is not efficient.

	Network graph	RC	CC of 1 [a]	CC of ℓ [a]
[KGSR02]	undirected	$O(n)$	$O(hn^2\rho)$	–
[DWB05] [b]	directed	1	$O(zn\rho)$	–
[YD09]	directed	quasi-poly. in z	quasi-poly. in $z\rho$	–
Our results	undirected	3 (Section 4.4.1)	$O(hn^2\rho)$	$O(h\ell\rho)$
		2 (Section 4.4.2)	$O(hn^2\rho)$	$O(hn\ell\rho)$
	directed	3 (Section 4.5.1)	$O(h^2n^2\rho)$	$O(hn\ell\rho)$
		2 (Section 4.5.2)	$O(h\rho)$	$O(h\ell\rho)$

[a] "CC of 1" is the CC of the PSMT protocol that transmits a single message and "CC of ℓ" is the CC of the protocol that transmits multiple (ℓ) messages, where each message is a field element of size ρ.

[b] Desmedt et al.'s protocol in [DWB05] is executed in special directed graphs without feedback paths, so only 1-round protocol is needed.

Table 4.1: PSMT in the general adversary model.

(see Section 3.2). First, the shares of an LSSS or the codewords of a linear code *are constructed with respect to the critical-path structure* $\mathcal{P}$, that is, *the participants of the secret sharing scheme are the critical paths in* P. Now we do not know the tight upper bound on h, but our Worst Case LSSS (see Section 3.2.1) achieves $h = O(|\mathcal{A}|n) = O(zn)$, so $h \leq zn$. In general, z is exponential in the size of the network graph. As we showed in Observation 3.3.1, the number of critical paths, n, can be linear or exponential in the size of the graph, depending on the network graph and the adversary structure. Thus we observe that z can be exponential in n, or n can be polynomial in z in some graphs (see also [KGSR02]). Either way, our protocols for transmitting a single message significantly improve the previous results in terms of CC and RC. We also present some efficient protocols to transmit $\ell > 1$ messages. The problem of multiple message transmission in the general adversary model has not been studied before.

The comparison of the results are shown in Table 4.1. Note that Desmedt et al.'s protocol in [DWB05] is executed in directed graphs without feedback, which means that the receiver R cannot send messages to the sender S. Thus the protocols in this graph must be non-interactive and can only have 1-round. Their protocol is actually an alternative use of our Worst Case LSSS that was presented earlier. Thus the protocol can easily be transformed into a 1-round protocol with CC $O(h\rho)$. The protocol by Yang and Desmedt [YD09] uses Patra et al.'s sub-protocol reconstruction scheme (see Appendix A.5), which requires both the CC and RC to be quasi-polynomial in z. As discussed above, both h and n are at most polynomial in z, so our improvements are obvious. We also remark that in the studies of the general adversary model, our results are the first to have constant RC in undirected and directed graphs with interaction.

4.3.2 Protocol Preliminaries

Recalling the statement at the beginning of Section 4.3, we employ the model of critical paths defined in Definition 3.3.2 and the critical-path structures defined in Definition 3.3.3. That is, we use $P = \{p_1, \ldots, p_n\}$, $Q = \{q_1, \ldots, q_u\}$, $\mathcal{P} = \{P_1, \ldots, P_z\}$ and $\mathcal{Q} = \{Q_1, \ldots, Q_z\}$ in our protocols.

We assume that each message m is drawn from the message space $\mathbb{M} \subseteq \mathbb{F}$ with respect to a certain probability distribution. Thus each message m is a field element of size ρ.

Given that the sender S and the receiver R are $2\mathcal{A}$-connected with the paths in P, if S sends the same elements via all paths in P, then R is able to recover these elements with perfect reliability [DWB05] (see Appendix A.4). In our protocols we say "*S broadcasts some elements via P*" to indicate this kind of transmission. Note that this *broadcast* only happens when S and R are $2\mathcal{A}$-connected, and the CC of the broadcast of 1 field element is $O(n\rho)$.

Our protocols use the linear code we described in Section 3.2. The participants of the linear code are the paths in P and Q, so *the linear codes are constructed with respect to the access structures $\Gamma = 2^P \setminus \mathcal{P}$ or $\Gamma' = 2^Q \setminus \mathcal{Q}$*. Thus, S usually sends a codeword $\mathbf{c} = \{c_1 \ldots, c_h\}$ via $P = \{p_1, \ldots, p_n\}$ in such a manner that for each $1 \leq j \leq h$, if $\psi(j) = p_i$ where $1 \leq i \leq n$, then S sends c_j via path p_i. We say "*S sends $\mathbf{c}$ via P with respect to ψ*" to indicate this kind of transmission. Note that the CC of the transmission of 1 codeword is $O(h\rho)$.

In addition, because the participants of the linear code are the critical paths, we use $sz^{\mathcal{P}}$ and $wt^{\mathcal{P}}$ to calculate the CC of the protocols, where $sz^{\mathcal{P}}$ and $wt^{\mathcal{P}}$ are alternations of $sz^{\mathcal{A}}$ and $wt^{\mathcal{A}}$ defined in Definition 3.2.5.

In our protocols, we omit some indices for the communication. For example, if S sends a pseudo-basis to R, then generally S should attach an index in the transmission to indicate exactly to which codeword each vector in the pseudo-basis corresponds. Indexing is very cheap in terms of CC. Thus in our protocols, we omit some indices to make the protocols easier to read.

In the following Section 4.4 and Section 4.5, we give our efficient PSMT protocols in undirected and directed graphs respectively. For each graph setting, we propose 3-round and 2-round protocols for the transmissions of a single message m and multiple (ℓ) messages $m_1, \ldots, m_\ell$. Thus we have four protocols in each section (for each graph setting).

4.4 Efficient PSMT in Undirected Graphs

In this section, we present our PSMT protocols in undirected graphs. The necessary and sufficient condition for PSMT in an undirected graph is that S and R are $2\mathcal{A}$-connected [KGSR02] (see Table 2.1). We first give 3-round protocols in Section 4.4.1 for the transmissions of a single message and multiple messages, and then give 2-round protocols in Section 4.4.2.

4.4.1 3-Round Undirected Protocols

First, we give a 3-round PSMT protocol for a single message transmission as follows.

3-Round Undirected Protocol for a single message m

Round 1 - S to R:

1. S chooses n random k-vectors $\mathbf{r}_1, \ldots, \mathbf{r}_n \in \mathbb{F}^k$, and for each $1 \leq i \leq n$, S encodes $\mathbf{r}_i$ to get codeword $\mathbf{c}_i = EC(\mathbf{r}_i) = (c_{i1}, \ldots, c_{ih})$.
2. For each $1 \leq i \leq n$, S sends vector $\mathbf{r}_i$ via path p_i, and sends codeword $\mathbf{c}_i$ via P with respect to ψ.

Round 2 - R to S:

1. R receives n k-vectors $\mathbf{r}'_1, \ldots, \mathbf{r}'_n$ and n h-vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ (regarding the codewords $\mathbf{c}_1, \ldots, \mathbf{c}_n$) from P. For each $1 \leq i \leq n$, let $\mathbf{x}_i = (x_{i1}, \ldots, x_{ih})$.
2. For each $1 \leq i \leq n$, R encodes $\mathbf{r}'_i$ to get codeword $\mathbf{c}'_i = EC(\mathbf{r}'_i) = (c'_{i1}, \ldots, c'_{ih})$. R then constructs a set D_i such that for each $1 \leq j \leq h$, if and only if $x_{ij} \neq c'_{ij}$, then $(x_{ij}, j) \in D_i$.
3. R broadcasts sets $D_1, \ldots, D_n$ via P.

Round 3 - S to R:

1. S receives sets $D_1, \ldots, D_n$ from P.
2. S sets $F := \emptyset$. For each $1 \leq i \leq n$, if there exists a pair $(x_{ij}, j) \in D_i$ such that $x_{ij} = c_{ij}$, then S sets $F := F \cup \{i\}$.
3. For each $1 \leq i \leq n$ such that $i \notin F$, S decodes $r_i = DC(\mathbf{r}_i)$. S computes $\sigma = m + \sum_{i \notin F} r_i$, and then broadcasts F and σ via P.

Recovery Phase

1. R receives F and σ from P.
2. For each $1 \leq i \leq n$ such that $i \notin F$, R decodes $r'_i = DC(\mathbf{r}'_i)$. R recovers $m' = \sigma - \sum_{i \notin F} r'_i$. **End.**

Theorem 4.4.1. *This 3-Round Undirected Protocol is a $(0,0)$-SMT protocol for a single message.*

Proof. Without loss of generality, we assume that the adversary corrupts the set of paths $\{p_1, \ldots, p_t\} \in \mathcal{P}$, and $c_{i1}, \ldots, c_{iy}$ are the entries in $\mathbf{c}_i$ that are assigned to these paths by ψ; i.e., $\psi(1, \ldots, y) = \{p_1, \ldots, p_t\}$.

First, we prove that the protocol is 0-private. In Round 1, the adversary can learn $\mathbf{r}_1, \ldots, \mathbf{r}_t$. We observe that if a pair $(x_{ij}, j) \in D_i$, then the adversary knows x_{ij} already before the broadcast of Round 2, because if $x_{ij} \neq c'_{ij}$, then either x_{ij} is changed, or $\mathbf{r}'_i$ is changed, or both are changed. In either case the adversary knows x_{ij}. Thus broadcasting D_i does not reveal extra information to the adversary. For each $i > t$, due to Lemma 3.2.1, with the entries $x_{i1}, \ldots, x_{iy}$ (i.e., $c_{i1}, \ldots, c_{iy}$) that the adversary can learn, the adversary cannot reconstruct $r_i = DC(\mathbf{r}_i)$. After Round 2, the adversary only knows $\mathbf{r}_1, \ldots, \mathbf{r}_t$ and hence $r_1, \ldots, r_t$. If $i > t$, then because $\mathbf{r}'_i = \mathbf{r}_i$, for each $1 \leq j \leq h$, we have $c'_{ij} = c_{ij}$. Thus $(x_{ij}, j) \in D_i \Rightarrow x_{ij} \neq c'_{ij} \Rightarrow x_{ij} \neq c_{ij}$, so $i \notin F$. Thus all the information $r_{t+1}, \ldots, r_n$ are used to encrypt the message m, so the adversary cannot recover m with only $r_1, \ldots, r_t$.

Next, we prove that the protocol is 0-reliable. We *claim* that if the adversary changes $\mathbf{r}_i$ on path p_i, then $i \in F$. Since S and R are $2\mathcal{A}$-connected, we have $\{p_{t+1}, \ldots, p_n\} \in \Gamma$.[6] If the adversary changes $\mathbf{r}_i$ to make $DC(\mathbf{r}'_i) \neq DC(\mathbf{r}_i)$, then there exists at least 1 entry c'_{ij}, where $\psi(j) \in \{p_{t+1}, \ldots, p_n\}$, in the codeword $EC(\mathbf{r}'_i) = (c'_{i1}, \ldots, c'_{ih})$ such that $x_{ij} \neq c'_{ij}$. This is because if such an entry does not exist, then due to Lemma 3.2.1, from the entries assigned to $\{p_{t+1}, \ldots, p_n\} \in \Gamma$, $DC(\mathbf{r}'_i) = DC(\mathbf{r}_i)$ can be recovered. Thus after Round 2, we have $(x_{ij}, j) \in D_i$ where $\psi(j) \in \{p_{t+1}, \ldots, p_n\}$. With this entry x_{ij}, S finds $x_{ij} = c_{ij}$, because $\{p_{t+1}, \ldots, p_n\}$ are uncorrupted, S adds i to F in Round 3. Thus we showed that our *claim* is correct. This claim implies that if $r'_i \neq r_i$, then $i \in F$. That is, for any $i \notin F$, we have $r'_i = r_i$. Thus R can recover $m' = \sigma - \sum_{i \notin F} r'_i = \sigma - \sum_{i \notin F} r_i$. This implies that the protocol is 0-reliable. □

CC of the protocol. Let $CC(i)$ be the CC of Round i for each $1 \leq i \leq 3$. In this protocol:[7]

$$\begin{aligned} CC(1) &= (kn + hn)\rho = O(hn\rho) \\ CC(2) &= O(2hn^2\rho) = O(hn^2\rho) \\ CC(3) &= O(n(sz^{\mathcal{P}} + 1)\rho) = O(n^2\rho) \end{aligned}$$

Therefore, the CC of this protocol is $O(hn^2\rho)$.

Next, we propose a 3-round PSMT protocol that transmits multiple ($\ell = wt^{\mathcal{P}}h$) messages. In this protocol we use the pseudo-basis that was discussed in Section 3.2.3.

3-Round Undirected Protocol for $\ell = wt^{\mathcal{P}}h$ messages $m_1, \ldots, m_\ell$

Round 1 - S to R:

1. S chooses $wt^{\mathcal{P}} + \ell$ random k-vectors $\mathbf{r}_1, \ldots, \mathbf{r}_{wt^{\mathcal{P}}+\ell} \in \mathbb{F}^k$, and for each $1 \leq i \leq wt^{\mathcal{P}} + \ell$, S encodes $\mathbf{r}_i$ to get codeword $\mathbf{c}_i = EC(\mathbf{r}_i) = (c_{i1}, \ldots, c_{ih})$.
2. For each $1 \leq i \leq wt^{\mathcal{P}} + \ell$, S sends codeword $\mathbf{c}_i$ via P with respect to ψ.

[6] Here $\Gamma = 2^P \setminus \mathcal{P}$, which is an access structure on the critical set of paths P.

[7] See the notations of $sz^{\mathcal{P}}$ and $wt^{\mathcal{P}}$ in Definition 3.2.5. Note that the linear code is constructed with respect to the critical-path structure $\mathcal{P}$. These notations will be used in the rest of this chapter.

Round 2 - R to S:

1. R receives $wt^{\mathcal{P}} + \ell$ h-vectors $\mathbf{x}_1, \ldots, \mathbf{x}_{wt^{\mathcal{P}}+\ell}$ (regarding the codewords $\mathbf{c}_1, \ldots, \mathbf{c}_{wt^{\mathcal{P}}+\ell}$) from P. For each $1 \leq i \leq wt^{\mathcal{P}} + \ell$, let $\mathbf{x}_i = (x_{i1}, \ldots, x_{ih})$.
2. R uses the Pseudo-Basis Construction Scheme (see Section 3.2.3) to construct a pseudo-basis B from $\mathbf{x}_1, \ldots, \mathbf{x}_{wt^{\mathcal{P}}+\ell}$, and then broadcasts B via P.

Round 3 - S to R:

1. S receives the pseudo-basis B from P.
2. Since the pseudo-dimension of B is at most $wt^{\mathcal{P}}$, S can find ℓ k-vectors $\mathbf{r}_{a_1}, \ldots, \mathbf{r}_{a_\ell}$ $(1 \leq a_1 < \ldots < a_\ell \leq wt^{\mathcal{P}} + \ell)$ such that $\mathbf{c}_{a_1}, \ldots, \mathbf{c}_{a_\ell} \notin B$. For each $1 \leq i \leq \ell$, S computes $r_{a_i} = DC(\mathbf{r}_{a_i})$ and $\sigma_i = m_i + r_{a_i}$.
3. S finds the final error locator F from B.[8] S then broadcasts F and $(\sigma_1, a_1), \ldots, (\sigma_\ell, a_\ell)$ via P.[9]

Recovery Phase

1. R receives F and $(\sigma_1, a_1), \ldots, (\sigma_\ell, a_\ell)$ from P.
2. With the final error locator F, R uses the Decoding Scheme from pseudo-basis (see Lemma 3.2.4) to get the information $r_{a_1}, \ldots, r_{a_\ell}$ of $\mathbf{c}_{a_1}, \ldots, \mathbf{c}_{a_\ell}$ from $\mathbf{x}_{a_1}, \ldots, \mathbf{x}_{a_\ell}$, and then for each $1 \leq i \leq \ell$, R recovers the message $m_i = \sigma_i - r_{a_i}$. **End.**

Theorem 4.4.2. *This 3-Round Undirected Protocol is a $(0,0)$-SMT protocol for multiple messages.*

Proof. First, the protocol is 0-private. This is because due to Lemma 3.2.1, the adversary cannot learn the information of any codeword in Round 1, and the codewords revealed in Round 2 are not used to encrypt the messages. Thus at the end of the protocol, the messages are encrypted with the information of the codewords that the adversary cannot learn, so 0-privacy is guaranteed. It is straightforward that the protocol is 0-reliable due to Lemma 3.2.4. □

CC of the protocol. In this protocol:

$$CC(1) = h(wt^{\mathcal{P}} + wt^{\mathcal{P}}h)\rho = O(wt^{\mathcal{P}}h^2\rho) = O(h\ell\rho)$$
$$CC(2) = O(n(wt^{\mathcal{P}}h)\rho) = O(n\ell\rho)$$
$$CC(3) = O(n(wt^{\mathcal{P}} + 2wt^{\mathcal{P}}h)\rho) = O(n\ell\rho)$$

Therefore, the CC of this protocol is $O(h\ell\rho)$.

[8] The final error locator F is a set of locations where the vectors in the pseudo-basis differ from the actual codewords. See Section 3.2.3 for the notation of the final error locator.

[9] $a_1, \ldots, a_\ell$ are used to indicate which vector $\mathbf{r}_{a_i}$ is used to compute σ_i, where $1 \leq i \leq \ell$.

4.4.2 2-Round Undirected Protocols

First, we give a 2-round PSMT protocol to transmit a single message (similar protocol in the threshold model has been given in [DESN10]).

2-Round Undirected Protocol for a single message m

Round 1 - R **to** S**:**

1. R chooses n random k-vectors $\mathbf{r}_1, \ldots, \mathbf{r}_n \in \mathbb{F}^k$, and for each $1 \leq i \leq n$, R encodes $\mathbf{r}_i$ to get codeword $\mathbf{c}_i = EC(\mathbf{r}_i) = (c_{i1}, \ldots, c_{ih})$.
2. For each $1 \leq i \leq n$, R sends vector $\mathbf{r}_i$ via path p_i, and sends codeword $\mathbf{c}_i$ via P with respect to ψ.

Round 2 - S **to** R**:**

1. S receives n k-vectors $\mathbf{r}'_1, \ldots, \mathbf{r}'_n$ and n h-vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ (regarding the codewords $\mathbf{c}_1, \ldots, \mathbf{c}_n$) from P. For each $1 \leq i \leq n$, let $\mathbf{x}_i = (x_{i1}, \ldots, x_{ih})$.
2. For each $1 \leq i \leq n$, S encodes $\mathbf{r}'_i$ to get codeword $\mathbf{c}'_i = EC(\mathbf{r}'_i) = (c'_{i1}, \ldots, c'_{ih})$. S then constructs a set D_i such that for each $1 \leq j \leq h$, if and only if $x_{ij} \neq c'_{ij}$, then $(x_{ij}, j) \in D_i$.
3. S finds a k-vector $\mathbf{r}^S$ such that $m = DC(\mathbf{r}^S)$, and then encodes $\mathbf{c}^S = EC(\mathbf{r}^S) = (c^S_1, \ldots, c^S_h)$. For each $1 \leq j \leq h$, if $\psi(j) = p_i$, then S computes $z_j = c^S_j + c'_{ij}$. Finally S sets $\mathbf{z} = (z_1, \ldots, z_h)$.
4. S broadcasts $\mathbf{z}$ and $D_1, \ldots, D_n$ via P.

Recovery Phase

1. R receives $\mathbf{z}$ and $D_1, \ldots, D_n$ from P.
2. R sets $F := \emptyset$. For each $1 \leq i \leq n$, if there exists a pair $(x_{ij}, j) \in D_i$ such that $x_{ij} = c_{ij}$, then R sets $F := F \cup \{i\}$.
3. For each $1 \leq j \leq h$, if $\psi(j) = p_i$, then R computes $c^R_j = z_j - c_{ij}$. R then decodes the message m' as the information of $(c^R_1, \ldots, c^R_h)$ such that for any $\psi(j) = p_i$ where $i \in F$, the entry c^R_j is not used for decoding. **End.**

Theorem 4.4.3. *This 2-Round Undirected Protocol is a* $(0, 0)$*-SMT protocol for a single message.*

Proof. Without loss of generality, we assume that the adversary corrupts the set of paths $\{p_1, \ldots, p_t\} \in \mathcal{P}$.

First, we prove that the protocol is 0-private. This follows the proof of Theorem 4.4.1. That is, the adversary can only learn $\mathbf{r}'_1, \ldots, \mathbf{r}'_t$. Thus in Round 2, the adversary can learn an entry $c^S_j = z_j - c'_{ij}$ in $\mathbf{c}^S$ if and only if $\psi(j) = p_i \in \{p_1, \ldots, p_t\}$. That is, the adversary can only learn the entries in $\mathbf{c}^S$ that are assigned to $\{p_1, \ldots, p_t\} \in \mathcal{P}$, and with these entries the adversary cannot recover the message m as the information of $\mathbf{c}^S$, due to Lemma 3.2.1.

Next, we prove that the protocol is 0-reliable. Following the proof of Theorem 4.4.1, we *claim* that if the adversary changes $\mathbf{r}_i$ on path p_i, then $i \in F$. Thus if $i \notin F$, then $\mathbf{r}'_i = \mathbf{r}_i$, and hence for each $1 \leq j \leq h$, we have $c'_{ij} = c_{ij}$. Thus if $i \notin F$, then $c^R_j = z_j - c_{ij} = z_j - c'_{ij} = c^S_j$. Due to the $2\mathcal{A}$-connectivity, with these correct entries in $(c^R_1, \ldots, c^R_h)$ (i.e., $c^R_j = c^S_j$), which are assigned to a set in the access structure $\Gamma = 2^P \backslash \mathcal{A}$, R can recover $m' = m$ with 0-reliability. This concludes the proof. □

CC of the protocol. In this protocol:

$$CC(1) = (kn + hn)\rho = O(hn\rho)$$
$$CC(2) = O(n(h + 2hn)\rho) = O(hn^2\rho)$$

Therefore, the CC of this protocol is $O(hn^2\rho)$.

Next, we show our 2-round PSMT protocol that transmits multiple messages. We employ a well-known technique in this context: the *randomness extractor* [SNR04, ACdH06, KS08] (see Appendix A.2). Suppose that the adversary has no knowledge on ℓ out of w random elements $r_1, \ldots, r_w \in \mathbb{F}$. Let $f(x)$ be a polynomial of degree $degf(x) = w - 1$ such that $f(i) = r_i$ for each $1 \leq i \leq w$, then the adversary has no knowledge on $z_j = f(w + j)$ for each $1 \leq j \leq \ell$. We denote a function $RE : \mathbb{F}^m \to \mathbb{F}^\ell$ as a randomness extractor such that $RE(r_1, \ldots, r_w) = (z_1, \ldots, z_\ell)$. This function will be used in the following 2-round PSMT protocol.

2-Round Undirected Protocol for $\ell = wt^{\mathcal{P}}(n - sz^{\mathcal{P}} - 1)$ messages $m_1, \ldots, m_\ell$

Round 1 - R to S:

1. R chooses $wt^{\mathcal{P}}n$ random k-vectors $\mathbf{r}_1, \ldots, \mathbf{r}_{wt^{\mathcal{P}}n} \in \mathbb{F}^k$, and for each $1 \leq i \leq wt^{\mathcal{P}}n$, S encodes $\mathbf{r}_i$ to get codeword $\mathbf{c}_i = EC(\mathbf{r}_i) = (c_{i1}, \ldots, c_{ih})$.
2. For each $1 \leq i \leq n$, R sends $wt^{\mathcal{P}}$ vectors

$$\mathbf{r}_{i+0\cdot n}, \mathbf{r}_{i+1\cdot n}, \ldots, \mathbf{r}_{i+(wt^{\mathcal{P}}-1)n}$$

via path p_i. R also sends codewords $\mathbf{c}_1, \ldots, \mathbf{c}_{wt^{\mathcal{P}}n}$ via P with respect to ψ.

Round 2 - S to R:

1. S receives $wt^{\mathcal{P}}$ k-vectors

$$\mathbf{r}'_{i+0\cdot n}, \mathbf{r}'_{i+1\cdot n}, \ldots, \mathbf{r}'_{i+(wt^{\mathcal{P}}-1)n}$$

on each path p_i $(1 \leq i \leq n)$, and also receives $wt^{\mathcal{P}}n$ h-vectors $\mathbf{x}_1, \ldots, \mathbf{x}_{wt^{\mathcal{P}}n}$ (regarding the codewords $\mathbf{c}_1, \ldots, \mathbf{c}_{wt^{\mathcal{P}}n}$) from P. For each $1 \leq i \leq wt^{\mathcal{P}}n$, let $\mathbf{x}_i = (x_{i1}, \ldots, x_{ih})$.

2. For each $1 \leq i \leq wt^{\mathcal{P}}n$, S uses the Pseudo-Basis Construction Scheme to construct a pseudo-basis B from $\mathbf{x}_1, \ldots, \mathbf{x}_{wt^{\mathcal{P}}n}$. Let b be the pseudo-dimension of B, then $b \leq wt^{\mathcal{P}}$.
3. For each $1 \leq i \leq wt^{\mathcal{P}}n$, S encodes $\mathbf{r}'_i$ to get codeword $\mathbf{c}'_i = EC(\mathbf{r}'_i) = (c'_{i1}, \ldots, c'_{ih})$. S then constructs a set D_i such that for each $1 \leq j \leq h$, if and only if $x_{ij} \neq c'_{ij}$, then $(c'_{ij}, x_{ij}, j) \in D_i$.
4. For each $1 \leq i \leq wt^{\mathcal{P}}n$, S decodes $r'_i = DC(\mathbf{r}'_i)$. S then constructs an ordered set T such that if and only if $|D_i| \leq wt^{\mathcal{P}}$, then $r'_i \in T$. S uses the randomness extractor to get $(z_1, \ldots, z_\ell) = RE(T)$, and for each $1 \leq i \leq \ell$, S computes $\sigma_i = m_i + z_i$.
5. S broadcasts the pseudo-basis B and $\sigma_1, \ldots, \sigma_\ell$ via P. For each $1 \leq i \leq wt^{\mathcal{P}}n$, if $|D_i| > wt^{\mathcal{P}}$, then S broadcasts "ignore i"; else, S broadcasts D_i via P.

Recovery Phase

1. R finds the final error locator F from B.
2. For each D_i that R receives on P, R constructs an h-vector $\mathbf{x}'_i = (x'_{i1}, \ldots, x'_{ih})$ such that for each $1 \leq j \leq h$, if $(c'_{ij}, x_{ij}, j) \in D_i$, then $x'_{ij} = c'_{ij} - (x_{ij} - c_{ij})$; else, then $x'_{ij} = c_{ij}$. R then decodes the information r''_i from $\mathbf{x}'_i$ such that for any $j \in F$, x'_{ij} is not used for decoding. R puts r''_i in an ordered set T'.
3. R uses the randomness extractor to get $(z'_1, \ldots, z'_\ell) = RE(T')$, and for each $1 \leq i \leq \ell$, R computes $m'_i = \sigma_i - z'_i$. **End.**

Theorem 4.4.4. *This 2-Round Undirected Protocol is a* $(0, 0)$*-SMT protocol for multiple messages.*

Proof. Without loss of generality, we assume that the adversary corrupts the set of paths $\{p_1, \ldots, p_t\} \in \mathcal{P}$.

First, we prove that the protocol is 0-private. In Round 1, the adversary can learn $wt^{\mathcal{P}}t$ random k-vectors:

$$\mathbf{r}'_{i+0\cdot n}, \mathbf{r}'_{i+1\cdot n}, \ldots, \mathbf{r}'_{i+(wt^{\mathcal{P}}-1)n}$$

for $1 \leq i \leq t$. With the pseudo-basis B broadcast in Round 2, the adversary can learn (at most) extra b codewords, and hence extra b random k-vectors. As shown the proof of Theorem 4.4.1, if a vector $(c'_{ij}, x_{ij}, j) \in D_i$, then the adversary knows c'_{ij} already before the broadcast of Round 2. That is, the broadcast in Round 2 does not reveal any extra information. Thus in total, the adversary can learn at most $wt^{\mathcal{P}}t + b$ $(\leq wt^{\mathcal{P}}(sz^{\mathcal{P}} + 1))$

random k-vectors that R has chosen in Round 1. Since

$$wt^{\mathcal{P}}n - (wt^{\mathcal{P}}t + b) \geq wt^{\mathcal{P}}(n - sz^{\mathcal{P}} - 1) = \ell,$$

there are at least ℓ k-vectors that remain secret. For any i such that the k-vector $\mathbf{r}_i$ remains secret, it is straightforward that $|D_i| \leq wt^{\mathcal{P}}$, and hence $r'_i \in T$ and r'_i is secret to the adversary. Thus the adversary has no knowledge on at least ℓ elements in T. S can then use the randomness extractor to get ℓ perfectly private random elements. That is, there are enough (at least ℓ) pads $z_1, \ldots, z_\ell$ to encrypt the messages, thus the protocol is 0-private.

Next, we prove that the protocol is 0-reliable. First we show that for each D_i that R receives, R gets $r''_i = r'_i$. First, for each $1 \leq i \leq wt^{\mathcal{P}}n$, we have $\mathbf{x}_i = \mathbf{c}_i + \mathbf{e}_i$ where $\mathbf{e}_i$ is an error vector. Due to Lemma 3.2.4, we know that the information of $\mathbf{c}_i$ can be decoded from $\mathbf{x}_i$ if the final error locator F is given. Let $\mathbf{e}_i = (e_{i1}, \ldots, e_{ih})$, for each $1 \leq j \leq h$, we have $x_{ij} = c_{ij} + e_{ij}$. Now in Recovery Phase, if $(c'_{ij}, x_{ij}, j) \in D_i$, then $x'_{ij} = c'_{ij} - (x_{ij} - c_{ij}) = c'_{ij} - e_{ij}$; else (which means $x_{ij} = c'_{ij}$), $x'_{ij} = c_{ij} = x_{ij} - e_{ij} = c'_{ij} - e_{ij}$. Thus in either case, for each $1 \leq j \leq h$, we have $x'_{ij} = c'_{ij} - e_{ij}$, and hence $\mathbf{x}'_i = \mathbf{c}'_i - \mathbf{e}_i$. Therefore, as we showed above, if the final error locator F is given, then the information of $\mathbf{c}'_i$ can be decoded from $\mathbf{x}'_i$. Thus R can get $r''_i = r'_i$ for each D_i received, and simultaneously get $(z'_1, \ldots, z'_\ell) = (z_1, \ldots, z_\ell)$ to recover the messages with 0-reliability. □

CC of the protocol. In this protocol:

$$CC(1) = (k + h)wt^{\mathcal{P}}n\rho = O(h\ell\rho)$$
$$CC(2) = O(n(wt^{\mathcal{P}}h + \ell + wt^{\mathcal{P}}n \cdot 3h)\rho) = O(h^2n^2\rho) = O(hn\ell\rho)$$

Therefore, the CC of this protocol is $O(hn\ell\rho)$.

4.5 Efficient PSMT in Directed Graphs

In this section we present our PSMT protocols in directed graphs. Again it is worth noting that the model of critical paths defined in Definition 3.3.2 and the critical-path structure defined in Definition 3.3.3 will be employed. That is, we use $P = \{p_1, \ldots, p_n\}$, $Q = \{q_1, \ldots, q_u\}$, $\mathcal{P} = \{P_1, \ldots, P_z\}$ and $\mathcal{Q} = \{Q_1, \ldots, Q_z\}$ in our protocols.

The necessary and sufficient condition for PSMT in a directed graph is that S and R are strongly $3\mathcal{A}$-directed-connected (see Definition 2.2.8). That is, S and R are $2\mathcal{A}$-connected on the forward paths (i.e., $P \notin 2\mathcal{P}$), and for any three sets $A_1, A_2, A_3 \in \mathcal{A}$, if $A_1 \cup A_2 \cup A_3$ cuts all the forward paths (i.e., $P_1 \cup P_2 \cup P_3 = P$), then at most one of these three sets cuts all the feedback paths (i.e., at most one $Q_i = Q$ where $i \in \{1, 2, 3\}$).

In a directed graph without feedback paths ($Q = \emptyset$), S and R are $3\mathcal{A}$-connected on

P. S only needs to send a codeword $\mathbf{c}$, of which the information is the message m, to R via P with respect to ψ. Due to Lemma 3.2.2, R can decode the information of $\mathbf{c}$ by correcting errors. Thus the protocol is perfectly secure and the CC is $O(h\rho)$. We note that Desmedt et al.'s protocol [DWB05] is actually an alternative use of the Worst Case LSSS (see Section 3.2.1).

In this section we consider a directed graph with feedback paths ($Q \neq \emptyset$). We give our 3-round protocols for single and multiple message transmission in Section 4.5.1. In Section 4.5.2, we show that the strong $3\mathcal{A}$-directed-connectivity is not sufficient for 2-round PSMT protocols in directed graphs, and hence we give a new necessary and sufficient condition and propose our protocols under this condition.

4.5.1 3-Round Directed Protocols

In our 3-round protocols, we do not need to assign shares (or entries) to the paths in Q. Thus the participants of the linear codes are the forward paths in P. That is, the linear code is constructed with respect to $\mathcal{P}$.

First, we give a 3-round PSMT protocol for the transmission of a single message.

3-Round Directed Protocol for a single message m

Round 1 - S **to** R**:**

1. S chooses $wt^{\mathcal{P}}(u+1)+1$ random k-vectors $\mathbf{r}_1, \ldots, \mathbf{r}_{wt^{\mathcal{P}}(u+1)+1} \in \mathbb{F}^k$, and for each $1 \leq i \leq wt^{\mathcal{P}}(u+1)+1$, S encodes $\mathbf{r}_i$ to get codeword $\mathbf{c}_i = EC(\mathbf{r}_i) = (c_{i1}, \ldots, c_{ih})$.
2. For each $1 \leq i \leq wt^{\mathcal{P}}(u+1)+1$, S sends $\mathbf{c}_i$ via P with respect to ψ.

Round 2 - R **to** S**:**

1. R receives $wt^{\mathcal{P}}(u+1)+1$ h-vectors $\mathbf{x}_1, \ldots, \mathbf{x}_{wt^{\mathcal{P}}(u+1)+1}$ (regarding the codewords $\mathbf{c}_1, \ldots, \mathbf{c}_{wt^{\mathcal{P}}(u+1)+1}$) from P. R uses the Pseudo-Basis Construction Scheme (see Section 3.2.3) to build a pseudo-basis B from $\mathbf{x}_1, \ldots, \mathbf{x}_{wt^{\mathcal{P}}(u+1)+1}$, and then sends B via all paths in Q.

Round 3 - S **to** R**:**

1. For each $1 \leq v \leq u$, let B_v be the pseudo-basis that S receives on path q_v, and let b_v be the pseudo-dimension of B_v.
2. For each $1 \leq v \leq u$, if $b_v > wt^{\mathcal{P}}$, then S broadcasts "ignore v" via P; else, S finds the final error locator F_v from B_v. If $|F_v| > wt^{\mathcal{P}}$, then S broadcasts "ignore v" via P; else, S broadcasts B_v and F_v via P.

3. S sets $U := \emptyset$ and $T := \emptyset$. For each $1 \leq v \leq u$ such that $b_v \leq wt^{\mathcal{P}}$ and $|F_v| \leq wt^{\mathcal{P}}$, S adds all the actual codewords ($\mathbf{c}_i$'s) that correspond to the h-vectors in B_v to U. Thus at last, $|U| \leq wt^{\mathcal{P}}u$. For each $\mathbf{r}_i$ such that $EC(\mathbf{r}_i) = \mathbf{c}_i \notin U$, if $i \notin T$ and $|T| < wt^{\mathcal{P}} + 1$, then S sets $T := T \cup \{i\}$. Thus at last, $|T| = wt^{\mathcal{P}} + 1$. For each $i \in T$, S decodes $r_i = DC(\mathbf{r}_i)$. S computes $\sigma = m + \sum_{i \in T} r_i$, and broadcasts σ and T via P.

Recovery Phase

Let $v := 1$, while $v \leq u$:

1. if R receives "ignore v" from P, then R sets $v := v + 1$;
2. else if R receives B_v and F_v from P, then
   (a) if $B_v \neq B$, then R sets $v := v + 1$;
   (b) else, with F_v, σ and T, R uses the Decoding Scheme from pseudo-basis (see Section 3.2.3) to get the information r_i of $\mathbf{c}_i$ for each $i \in T$. R then recovers $m = \sigma - \sum_{i \in T} r_i$, and terminates the protocol.

If $v > u$, then R knows that S did not receive the correct pseudo-basis B, so all paths in Q are corrupted. For each $i \in T$, R finds a set $P_f \in \mathcal{P}$ such that $Q_f = Q$, and if P_f's entries in $\mathbf{x}_i$ are removed, all the remaining entries are a part of a codeword $\mathbf{c}'_i \in C$, R then decodes r'_i as the information of $\mathbf{c}'_i$. R recovers $m' = \sigma - \sum_{i \in T} r'_i$. **End.**

Theorem 4.5.1. *This 3-Round Directed Protocol is a* $(0, 0)$*-SMT protocol for a single message.*

Proof. First, we prove that the protocol is 0-private. There are in total $wt^{\mathcal{P}}(u + 1) + 1$ codewords being transmitted. In Round 1, the adversary cannot learn any codeword. In Round 2, the adversary can learn at most b codewords from the pseudo-basis B that R sends, where b is the pseudo-dimension of B. On each path $q_v \in Q$, the adversary can change the pseudo-basis to B_v so that in Round 3, when S broadcasts F_v, the adversary can learn if the codewords corresponding to the guessed h-vectors in B_v are correct. This is a kind of *Guessing Attack* on the feedback paths (see Section 4.1). The pseudo-dimension b_v must not be larger than $wt^{\mathcal{P}}$ for the Guessing Attack to be successful, because otherwise S and R will "ignore v". Thus the set U that S constructs in Round 3 consists of all the codewords that the adversary can possibly learn by performing the Guessing Attack. As mentioned earlier, besides these $|U| \leq wt^{\mathcal{P}}u$ codewords, the adversary can learn at most b other codewords from the pseudo-basis B, which are sent by R in Round 2. Since $b \leq wt^{\mathcal{P}}$, the adversary can learn at most $|U| + b \leq wt^{\mathcal{P}}(u + 1)$ codewords. Thus it cannot learn the information of at least 1 out of the $wt^{\mathcal{P}} + 1$ codewords that are not in U and indicated by T. Since this secret information is used to encrypt the message m, the adversary cannot recover m, which means that the protocol is 0-private.

Next we prove that the protocol is 0-reliable. If S receives some $B_v = B$ on a path $q_v \in Q$, then S will send B_v and F_v to R in Round 3. Thus it is straightforward that R can recover the message (Recovery Phase, case 1(b)) due to Lemma 3.2.4. Otherwise, if S does not receive a correct B, then in Recovery Phase, R is able to know that all paths in Q are corrupted. R will then perform the last part of our protocol. We *claim* that for each r'_i that R decodes, we have $r'_i = r_i$. Let $P_e \in \mathcal{P}$ be the set that the adversary corrupts, so $Q_e = Q$. Assume that there is an $i \in T$ such that $r'_i \neq r_i$, then some of P_e's entries are in the remaining entries of $\mathbf{x}_i$ after removing P_f's entries. Now the remaining entries are a part of a codeword means that there exists a set $P_j \in \mathcal{P}$, such that all the remaining entries are assigned to $P_e \cup P_j$ by ψ. This is because if P_j does not exist, then the remaining entries that are not assigned to P_e are assigned to a set in the access structure $\Gamma = 2^P \setminus \mathcal{P}$. With these entries, $r'_i = r_i$ can be decoded, due to Lemma 3.2.1. Thus P_j exists, so to sum up we have $P_f \cup P_e \cup P_j = P$, $Q_f = Q$ and $Q_e = Q$. This contradicts the strong $3\mathcal{A}$-directed-connectivity. From this contradiction, we showed that our *claim* is correct. Thus with the correct $r'_i = r_i$ for each $i \in T$, R can recover the message m with 0-reliability. □

CC of the protocol. In this protocol:

$$\begin{aligned}
CC(1) &= h(wt^{\mathcal{P}}(u+1)+1)\rho = O(h^2 n\rho) \\
CC(2) &= O(u(wt^{\mathcal{P}}h)\rho) = O(h^2 n\rho) \\
CC(3) &= O(n(wt^{\mathcal{P}}hu + wt^{\mathcal{P}}u + 1 + (wt^{\mathcal{P}}+1))\rho) = O(h^2 n^2 \rho)
\end{aligned}$$

Therefore, the CC of this protocol is $O(h^2 n^2 \rho)$.

Next, we propose a 3-round PSMT protocol that transmits multiple ($\ell = wt^{\mathcal{P}}u$) messages. This protocol is adapted from the above protocol which transmits a single message. Thus we only show their differences as follows.

3-Round Directed Protocol for $\ell = wt^{\mathcal{P}}u$ message $m_1, \ldots, m_\ell$

Round 1 - S to R: S does the same only for $wt^{\mathcal{P}}(u+1)+\ell$ random k-vectors.

Round 2 - R to S: R does the same.

Round 3 - S to R: S does the same until step 3.

3. S sets $U := \emptyset$. For each $1 \leq v \leq u$ such that $b_v \leq wt^{\mathcal{P}}$ and $|F_v| \leq wt^{\mathcal{P}}$, S adds all the actual codewords ($\mathbf{c}_i$'s) that correspond to the h-vectors in B_v to U. Thus at last, $|U| \leq wt^{\mathcal{P}}u$.
4. S sets $T_1, \ldots, T_\ell := \emptyset$. For each $\mathbf{r}_i$ such that $EC(\mathbf{r}_i) = \mathbf{c}_i \notin U$, for each $1 \leq j \leq \ell$, if $i \notin T_j$ and $|T_j| < wt^{\mathcal{P}}$, then S sets $T_j := T_j \cup \{i\}$.

Thus all $T_1, \ldots, T_\ell$ are *the same* and each $|T_j| = wt^{\mathcal{P}}$. There are at least ℓ k-vectors $\mathbf{r}_i$ such that $EC(\mathbf{r}_i) = \mathbf{c}_i \notin U$ and $i \notin T_j$.[10] Let $\mathbf{r}_{i_1}, \ldots, \mathbf{r}_{i_\ell}$ be ℓ such k-vectors, then for each $1 \leq j \leq \ell$, S sets $T_j := T_j \cup \{i_j\}$. Thus $|T_j| = wt^{\mathcal{P}}+1$, and all $T_1, \ldots, T_\ell$ are *different.* For each $1 \leq j \leq \ell$ and $i \in T_j$, S decodes $r_i = DC(\mathbf{r}_i)$, computes $\sigma_j = m_j + \sum_{i \in T_j} r_i$, and broadcasts σ_j and T_j via P.

Recovery Phase For each $1 \leq j \leq \ell$, R does the same to recover the message m_j. **End.**

Theorem 4.5.2. *This 3-Round Directed Protocol is a* $(0,0)$*-SMT protocol for multiple messages.*

Proof. The (0,0)-security of this protocol can be easily proved following the proof of Theorem 4.5.1. □

CC of the protocol. In this protocol:

$$\begin{aligned}
CC(1) &= h(wt^{\mathcal{P}}(u+1) + wt^{\mathcal{P}}u)\rho = O(h\ell\rho) \\
CC(2) &= O(u(wt^{\mathcal{P}}h)\rho) = O(h\ell\rho) \\
CC(3) &= O(n(wt^{\mathcal{P}}hu + wt^{\mathcal{P}}u + wt^{\mathcal{P}}u(1 + (wt^{\mathcal{P}}+1)))\rho) = O(hn\ell\rho)
\end{aligned}$$

Therefore, the CC of this protocol is $O(hn\ell\rho)$.

4.5.2 2-Round Directed Protocols

In [PCR09], Patra et al. showed that in the threshold model, the minimal connectivity for PSMT in directed graphs (i.e., $n \geq \max\{3t+1-2u, 2t+1\}$) is not sufficient for 2-round protocols. Here we show a similar result in the general adversary model. That is, we prove that in the general adversary model, the strong $3\mathcal{A}$-directed-connectivity is not sufficient for 2-round protocols. Note that the general assumption is that the feedback paths are not reliable.

Theorem 4.5.3. *Given a directed graph* $G(V,E)$ *where* $S, R \in V$*, let* $\mathcal{A}$ *be an adversary structure on* $V \setminus \{S,R\}$*, 2-round PSMT is possible if and only if* S *and* R *are* $2\mathcal{A}$*-connected on the forward paths and* $3\mathcal{A}$*-connected with the union of all the forward and feedback paths in* G*.*

Proof. First we prove the necessity of the condition. The $2\mathcal{A}$-connectivity on the forward paths is obviously necessary. Now for contradiction, we assume that S and R are $3\mathcal{A}$-separated in G and a 2-round PSMT protocol Π exists. Let $view^S$ and $view^R$ be the

[10] Because $|U| \leq wt^{\mathcal{P}}u$, $|T_j| = wt^{\mathcal{P}}$ and the total number of vectors $\mathbf{r}_i$ is $wt^{\mathcal{P}}(u+1)+\ell$, it is straightforward that the number of vectors $\mathbf{r}_i$ such that $EC(\mathbf{r}_i) = \mathbf{c}_i \notin U$ and $i \notin T_j$ is $wt^{\mathcal{P}}(u+1) + \ell - |U| - |T_j| \geq \ell$.

views of S and R respectively. In Round 1 of Π, $view^S$ and $view^R$ can be different if the adversary corrupts some feedback paths. Since the feedback paths are not reliable, S cannot detect the differences. Thus after Round 2, because Π is perfectly private, with respect to $\mathcal{P} \diamond \mathcal{Q}$ (see Definition 3.3.3), we regard $view^S$ as a codeword whose information is the message. Thus $view^R$ is $view^S$ plus an error vector caused by a corrupted set $P_e \cup Q_e \in \mathcal{P} \diamond \mathcal{Q}$. Since S and R are $3\mathcal{A}$-separated, we have $P \cup Q \in 3(\mathcal{P} \diamond \mathcal{Q})$. Thus due to Lemma 3.2.2, R cannot correct the errors and hence cannot decode the message. Thus Π is not perfectly reliable. This contradiction shows that the condition is necessary.

Next we show a 2-round PSMT protocol under this condition. First we consider the critical-path structure $\mathcal{P} \diamond \mathcal{Q}$. From the condition, we have $P \notin 2\mathcal{P}$ and $P \cup Q \notin 3(\mathcal{P} \diamond \mathcal{Q})$. Now if $Q \notin \mathcal{P} \diamond \mathcal{Q}$, then we add Q to $\mathcal{P} \diamond \mathcal{Q}$. It is straightforward that with this updated critical-path structure $\mathcal{P} \diamond \mathcal{Q}$, we still have $P \notin 2\mathcal{P}$ and $P \cup Q \notin 3(\mathcal{P} \diamond \mathcal{Q})$. In the following protocol, we consider that paths in $P \cup Q$ are the participants of our linear code, thus the linear code in this protocol is constructed with respect to the updated $\mathcal{P} \diamond \mathcal{Q}$. Since $Q \in \mathcal{P} \diamond \mathcal{Q}$, there exists a linear code defined by a generating matrix G' (see Definition 3.2.3) such that the columns in G' that are assigned to the paths in Q are linearly independent. We use this linear code in our protocol.

2-Round Directed Protocol for a single message m

Round 1 - R to S: R chooses a random k-vector $\mathbf{r}$, and encodes it to get the codeword $\mathbf{c} = EC(\mathbf{r}) = (c_1, \dots, c_h)$. Suppose that $c_1, \dots, c_t$ are the entries in $\mathbf{c}$ such that $\psi(c_1, \dots, c_t) = Q$, our linear code allows all these entries to be independent. R then sends the entries $c_1, \dots, c_t$ via Q with respect to ψ.

Round 2 - S to R: Upon the entries $c'_1, \dots, c'_t$ that S receives on Q, S constructs a k-vector $\mathbf{r}'$ such that $c'_1, \dots, c'_t$ are a part of the codeword $\mathbf{c}' = EC(\mathbf{r}') = (c'_1, \dots, c'_t, c'_{t+1}, \dots, c'_h)$. S decodes $r' = DC(\mathbf{r}')$. S then sends $c'_{t+1}, \dots, c'_h$ via P with respect to ψ and broadcasts $\sigma = m + r'$ via P.

Recovery Phase R receives $c''_{t+1}, \dots, c''_h$ and σ on P. R constructs an h-vector $\mathbf{x} = (c_1, \dots, c_t, c''_{t+1}, \dots, c''_h)$. Thus $\mathbf{x} = \mathbf{c}' + \mathbf{e}$ where $\mathbf{e}$ is an error vector caused by a corrupted set $P_e \cup Q_e \in \mathcal{P} \diamond \mathcal{Q}$ where $1 \leq e \leq z$. Since S and R are $3\mathcal{A}$-connected, due to Lemma 3.2.2, R can correct the errors, decode the information r' of $\mathbf{c}'$ from $\mathbf{x}$, and recover the message $m = \sigma - r'$. **End.**

We now prove the security of this protocol. First, the protocol is 0-private. Because the codeword $\mathbf{c}'$ is transmitted via $P \cup Q$ with respect to ψ. Due to Lemma 3.2.1, the adversary cannot decode r' and hence cannot recover m. Next, it is straightforward that the protocol is 0-reliable due to Lemma 3.2.2.

Clearly the CC of this 2-Round Directed Protocol is $O(h\rho)$. Furthermore, this protocol can be used to transmit any $\ell > 1$ messages with CC $O(h\ell\rho)$. □

4.6 Brief Conclusion of Chapter 4

In this chapter, we presented our results on SMT in point-to-point networks. First in Section 4.1, we proposed a Guessing Attack on some existing PSMT protocols in a directed graph with feedback paths. Our Guessing Attack was described using two examples: one in the threshold model and the other in the general adversary model. Next in Section 4.2, we determined the minimal network connectivities for APSMT. We showed the necessary and sufficient conditions for δ-RMT and $(0, \delta)$-SMT in both undirected and directed graphs, and then proved that the minimal connectivity required for ϵ-privacy is the same as that for 0-privacy. These results now complete Table 2.1, and the completed table will be shown in Section 6.1. In Section 4.4 and Section 4.5, we gave efficient PSMT protocols in undirected and directed graphs respectively. The constructions of our protocols are based on the ideas of linear code, pseudo-basis and critical paths. These protocols make significant improvements to the previous results in terms of communication complexity (CC) and round complexity (RC) (see the comparison of the results in Section 4.3).

In the next chapter, we study SMT in multicast communication neighbour networks. Our goal is to determine the minimal connectivities for SMT in a general multicast graph in both the threshold and general adversary models.

Chapter 5

SMT in Multicast Networks

In this chapter we solve the problem of secure message transmission (SMT) in multicast graphs. A multicast graph is an undirected graph used to model the neighbour network (see Section 2.2.2), in which a message multicast by a node is received—simultaneously and privately—by all its neighbours. Our work is devoted to determine minimal connectivities for different levels of reliability and security in a general multicast graph, in which node-disjoint and neighbour-disjoint paths are not strictly required. Thus our results solve Franklin and Wright's open problem [FW98] (see Section 1.3), which has been open for over a decade.

In our work, we study SMT in multicast graphs in the general adversary model, and then apply the results to the threshold model.

First, we show that the current Basic Characterization (see Definition 3.3.2) of the network graph is not enough to characterize the multicast communication. Thus in Section 5.1, we give an *Extended Characterization* of the multicast graphs, which is based on our observation on the *eavesdropping* and *separating* activities of the adversary on a single path. This characterization should give a clearer insight on how the message can be securely transmitted over multicast graphs.

Next, in Section 5.2 and Section 5.3, we give the necessary and sufficient conditions for reliability and security respectively. Besides proving that our conditions imply the lower bounds on network connectivity, we also provide message transmission protocols to show that these bounds are tight.

Finally in Section 5.4, we use our results in the general adversary model to find the necessary and sufficient conditions for reliability and security in the threshold model. Moreover, by analysing the previous results, we show how our results explain all the examples and prove all the conjectures in the previous studies (see some previous results in Section 2.9.2).

Some results in this chapter has been published in [YD11].

5.1 Characterization of Multicast Communication

In this section we characterize the multicast graphs based on the adversary structures. We show an *Extended Characterization* which is essential for obtaining the necessary and sufficient conditions in the multicast model. This characterization is based on our observation on the *eavesdropping* and *separating* activities of the adversary on a single path between the sender S and the receiver R. We show this observation in the following Section 5.1.1, and present the Extended Characterization in Section 5.1.2.

5.1.1 Eavesdropping and Separating

Given an undirected multicast graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$ and P be the set of all paths between S and R. For each path $p \in P$, we define *eavesdropping* and *separating* as follows.

Definition 5.1.1. *We say that the adversary can* eavesdrop on *p if it* cannot *control any node on p but can control some neighbours of p.*[1] *Suppose that the adversary can eavesdrop on p and there is an element a to be transmitted between S and R on p. We say that the adversary can* completely eavesdrop on *p if the adversary can always learn a by eavesdropping.*

Definition 5.1.2. *We say that the adversary can* separate S and R on p *if it can control some nodes on p. Suppose that the adversary can separate S and R on p and there are k elements* $(a_1, \ldots, a_k) \in \mathbb{F}^k$ *to be transmitted on p. We let* $(a_1^S, \ldots, a_k^S)$ *and* $(a_1^R, \ldots, a_k^R)$ *be the views of S and R respectively on these k elements at the end of any protocol. We say that the adversary can* completely separate S and R on p *if there always exists a strategy of the adversary that causes* $\forall i\ (1 \leq i \leq k) : a_i^S \neq a_i^R$ *with probability 1.*

Next we show several lemmas regarding the eavesdropping and separating activities of the adversary on a single path $p \in P$. In the following, we assume that path p is placed in a *left-to-right* direction, with S at the left end and R at the right end.

Lemma 5.1.1. *The adversary can completely eavesdrop on a path* $p \in P$ *if and only if it can eavesdrop on two adjacent nodes*[2] *on p.*

Proof. We first prove the "if" direction. The privacy problem has been studied by Franklin and Yung in [FY95]. They showed that private communication is possible only if S and R are weakly t_{hyper}-connected in the hypergraph $H_G(V, E_{H_G})$ (see Definition 2.2.4). Concerning private communication on a path p, this connectivity means that by removing all the faulty nodes and the hyperedges on which the faulty nodes are, path p remains. Evidently, this necessary condition for privacy is satisfied if and only

[1] Obviously, if the adversary *can* control some nodes on p, then it can learn everything passing through those controlled nodes. However, for the purpose of our observation, we do not consider this activity as "eavesdropping", instead, we characterize it as "separating", which we describe in Definition 5.1.2.

[2] Two nodes $u, v \in V$ are *adjacent* to one another if there is an edge $\{u, v\} \in E$ between them.

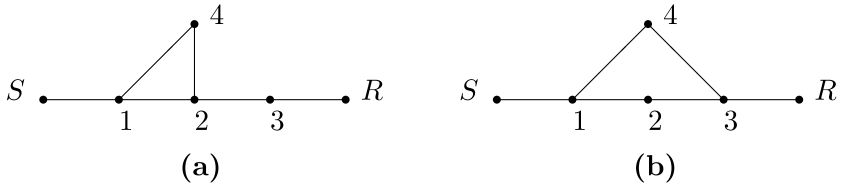


Figure 5.1: Eavesdropping activities on a single path p.

if the adversary *cannot* eavesdrop on two adjacent nodes on p. This is clear from our Single Path Eavesdropping Examples following this proof. Thus if the adversary can eavesdrop on two adjacent nodes on p, then it can completely eavesdrop on p.

Next we prove the "only if" direction. We propose the following protocol, which allows S to send an element a^S to R with perfect privacy, if the adversary cannot eavesdrop on two adjacent nodes on p. First we assume that including S and R, there are $k+2$ nodes $v_0, \ldots, v_{k+1}$ on p. We let S be node v_0, R be node v_{k+1}, and $v_1, \ldots, v_k$ be the other k nodes from left to right.

Single Path Private Propagation Protocol

1. For each $1 \leq i \leq k+1$, v_i initiates an element $a_i \in_R \mathbb{F}$ and multicasts it. Thus for each $0 \leq i \leq k$, v_i receives the element a_{i+1} from its right side neighbour node v_{i+1}.
2. S sets $i := 1$ and multicasts $b_0 = a^S + a_1$. While $i \leq k$, v_i receives the element b_{i-1} from its left side neighbour node v_{i-1}, v_i then multicasts $b_i = b_{i-1} - a_i + a_{i+1}$ and sets $i := i+1$.
3. When $i = k+1$, R receives the element b_k from v_k, R then sets $a^R := b_k - a_{k+1}$. **End.**

Now for each $0 \leq i \leq k$, the element that v_i multicasts is an encrypted ciphertext $b_i = a^S + a_{i+1}$. In order to decrypt a^S, the adversary needs to learn a pair (b_i, a_{i+1}) for some $0 \leq i \leq k$. Since b_i is multicast by v_i and a_{i+1} is multicast by v_{i+1}, the adversary who cannot eavesdrop on two adjacent nodes is not able to learn a^S by eavesdropping. □

Single Path Eavesdropping Examples

(a) If the adversary can eavesdrop on at least two adjacent nodes on p, then the necessary condition of [FY95] is not satisfied. For example, in Fig 5.1(a), the faulty node is node 4 and the hyperedges are

$$(S, \{1\}), (1, \{S, 2, 4\}), (2, \{1, 3, 4\}), (3, \{2, R\}), (4, \{1, 2\}) \text{ and } (R, \{3\}).$$

By removing the hyperedges that node 4 is on, the remaining hyperedges are

$$(S, \{1\}), (3, \{2, R\}) \text{ and } (R, \{3\}).$$

Thus p does not remain because edge $\{1,2\}$ is removed, and hence the condition of [FY95] is not satisfied.

(b) If the adversary cannot eavesdrop on two adjacent nodes on p, then the necessary condition of [FY95] is satisfied. For example, in Fig 5.1(b), the faulty node is node 4 and the hyperedges are

$$(S,\{1\}),(1,\{S,2,4\}),(2,\{1,3\}),(3,\{2,4,R\}),(4,\{1,3\}) \text{ and } (R,\{3\}).$$

By removing the hyperedges that node 4 is on, the remaining hyperedges are

$$(S,\{1\}),(2,\{1,3\}) \text{ and } (R,\{3\}).$$

Thus p remains because all edges on p remain, and hence the condition of [FY95] is satisfied.

We note that the separating activities have been observed by Franklin and Wright in [FW98], but here we extend their result and upgrade their protocol.

Lemma 5.1.2. (following [FW98]) *The adversary can completely separate S and R on a path $p \in P$ if and only if it can control two or more nodes on p.*

Proof. We first prove the "if" direction. If the adversary can control two or more nodes on p, then the leftmost faulty node, say v_1, will modify whatever is received on its right side, and the rightmost faulty node, say v_2, will modify whatever is received on its left side. Thus despite what protocol is executed, in the view of S, the elements transmitted on the right side of v_1 are corrupted, and in the view of R, the elements transmitted on the left side of v_2 are corrupted. This immediately implies that the views of S and R can be completely different.

Next we prove the "only if" direction. We assume that including S and R, there are $k+2$ nodes $v_0,\ldots,v_{k+1}$ on p. We let S be node v_0, R be node v_{k+1}, and $v_1,\ldots,v_k$ be the other k nodes from left to right. We show that with the following protocol, the adversary cannot completely separate S and R when k elements $(a_1,\ldots,a_k)$ are transmitted on p, if the adversary can only control one node on p.

Single Path Distribution Protocol

1. For each $1 \le i \le k$, v_i initiates an element $a_i \in_R \mathbb{F}$ and multicasts it.
2. For each $1 \le i \le k$, the nodes on the left side of v_i execute an instance of the Single Path Private Propagation Protocol from v_{i-1} to S in which v_{i-1} sends a_i, and the nodes on the right side of v_i execute an instance of the Single Path Private Propagation Protocol from v_{i+1} to R in which v_{i+1} sends a_i.

3. At the end of the protocol, for each $1 \leq i \leq k$, S receives an element a_i^S and R receives an element a_i^R. If S (or R) receives nothing regarding the element a_i for some $1 \leq i \leq k$, then S (or R) sets $a_i^S = 1$ (or $a_i^R = 1$). **End.**

Let v_e ($1 \leq e \leq k$) be the only faulty node on p. It is straightforward that at the end of the protocol, $a_e^S = a_e^R$, even if v_e does not initiate and multicast any element (in this case $a_e^S = a_e^R = 1$). This shows that the adversary cannot completely separate S and R on path p if it can only control one node on p. □

Finally, we give the following two lemmas, which are trivial so we omit the proofs.

Lemma 5.1.3. *If the adversary can only control one node v on a path $p \in P$, then despite what protocol is executed on p, there exists a strategy of the adversary that causes the views of S and R to be different except for their views on the elements multicast by v.*

Lemma 5.1.4. *Given a node v on a path $p \in P$, if the adversary can neither separate S and R on p, nor completely eavesdrop on p, nor control a neighbour of v, then during the execution of the Single Path Distribution Protocol on p, the adversary cannot learn the elements multicast by v.*

5.1.2 Extended Characterization

Based on our observation on the eavesdropping and separating activities, we now present an *Extended Characterization* $\zeta_{\mathcal{A}}$ of a multicast graph $G(V, E)$ given an adversary structure $\mathcal{A}$ on $V \setminus \{S, R\}$. This is an extension of the Basic Characterization shown in Definition 3.3.2. This Extended Characterization is essential for obtaining the necessary and sufficient conditions in multicast graphs.

Definition 5.1.3. *Given an undirected multicast graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A} = \{A_1, \ldots, A_z\}$ be an adversary structure on $V \setminus \{S, R\}$ and P be the set of all paths between S and R. An* Extended Characterization *of G given $\mathcal{A}$ is a set $\zeta_{\mathcal{A}} = \{\zeta_{A_1}, \ldots, \zeta_{A_z}\}$ where for each $1 \leq i \leq z$, we have $\zeta_{A_i} = (P_i^{(+)}, P_i^{(1)}, P_i^{(*)}, P_i)$ where*

- *$P_i^{(+)}$ is the set of all paths on each of which there are at least two nodes in A_i,*
- *$P_i^{(1)}$ is the set of all paths on each of which there is exactly one node in A_i,*
- *$P_i^{(*)}$ is the set of all paths on each of which there is no node in A_i, but on each path in $P_i^{(*)}$, there are two adjacent nodes that both have neighbours in A_i, and*
- *$P_i = P_i^{(+)} \cup P_i^{(1)}$ is the set of all paths that A_i cuts (same as Definition 3.3.2).*

With the Extended Characterization $\zeta_{\mathcal{A}}$, we know that during the execution of any protocol, by choosing a set $A_i \in \mathcal{A}$ to control, the adversary can separate S and R on P_i, completely separate S and R on $P_i^{(+)}$ and completely eavesdrop on $P_i^{(*)}$.

Given any set $A_i \in \mathcal{A}$, we are particularly interested in the nodes of A_i on the paths of $P_i^{(1)}$. Due to Definition 5.1.3, there is exactly one node in A_i on each path in $P_i^{(1)}$. For each path $p \in P_i^{(1)}$, *we use $A_i \sqcap p$ to denote the single node $v \in A_i$ that is on path p*; i.e., $v = A_i \sqcap p$. Note that this notation is only used for the paths in $P_i^{(1)}$, but not for $P_i^{(+)}$.

Finally in this section, we define some special connectivities (e.g., *high $\mathcal{A}$-connectivity* and *low $2\mathcal{A}$-connectivity*) using the Extended Characterization.

Definition 5.1.4. *Given a graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, we say that S and R are* highly $\mathcal{A}$-connected *if for any set $A_i \in \mathcal{A}$, we have $P_i \cup P_i^{(*)} \neq P$.*

Definition 5.1.5. *Given a graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, we say that S and R are* highly $2\mathcal{A}$-connected *if for any two sets $A_i, A_j \in \mathcal{A}$, we have $(P_i \cup P_i^{(*)}) \cup P_j \neq P$.*

Definition 5.1.6. *Given a graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, we say that S and R are* lowly $2\mathcal{A}$-separated *if there exist two (not necessarily distinct) sets $A_1, A_2 \in \mathcal{A}$ such that*

(a) $P_1 \cup P_2 = P$, and

(b) $P_1^{(1)} = \emptyset$, or for each path $p \in P_1^{(1)}$, we have that $p \in P_2 \cup P_2^{(*)}$ or $A_1 \sqcap p$ has a neighbour in A_2, and

(c) $P_2^{(1)} = \emptyset$, or for each path $p \in P_2^{(1)}$, we have that $p \in P_1 \cup P_1^{(*)}$ or $A_2 \sqcap p$ has a neighbour in A_1.

We say that S and R are lowly $2\mathcal{A}$-connected *if they are not lowly $2\mathcal{A}$-separated.*

Lemma 5.1.5. *Given a graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, if S and R are lowly $2\mathcal{A}$-connected, then they are $\mathcal{A}$-connected.*

Proof. Assume that S and R are $\mathcal{A}$-separated; i.e., there exits a set $A_i \in \mathcal{A}$ such that $P_i = P$. If we let both the sets A_1, A_2 of Definition 5.1.6 be A_i, then it is straightforward that S and R are lowly $2\mathcal{A}$-separated. Thus we have a contradiction. □

With some examples, we can show that the high $\mathcal{A}$-connectivity and the low $2\mathcal{A}$-connectivity do not imply each other. We give more details in Section 5.4.

Next, we present our results on reliable communication in Section 5.2, and then study secure communication in Section 5.3.

5.2 Reliable Multicast Communication

In this section, we discuss reliable message transmission (RMT) in multicast graphs. We study almost perfect reliability (δ-RMT) in Section 5.2.1 and perfect reliability (0-RMT) in Section 5.2.2.

5.2.1 Almost Perfectly Reliable Multicast

Here we give the necessary and sufficient condition for δ-RMT in a multicast graph.

Theorem 5.2.1. *Given a graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, the necessary and sufficient condition for δ-RMT from S to R is that S and R are lowly $2\mathcal{A}$-connected.*

Next, we use Lemma 5.2.2 to show that the condition is necessary and Lemma 5.2.3 to show that the condition is sufficient. Before we present these two lemmas, we first give the following Lemma 5.2.1, which is a key ingredient for proving the necessity.

Lemma 5.2.1. *Given a graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, if there exists two sets $A_1, A_2 \in \mathcal{A}$ such that $P_1^{(+)} \cup P_2^{(+)} = P$, and $\delta < \frac{1}{2}(1 - \frac{1}{|\mathbb{M}|})$, then δ-RMT from S to R is impossible.*

Proof. First, we note that similar results has been proven in [FW00, Theorem 5.1] and [DWB05, Theorem 3] in the point-to-point setting. Especially in [FW00], Franklin and Wright showed that if $n \leq 2t$ and $\delta < \frac{1}{2}(1 - \frac{1}{|\mathbb{M}|})$, then in a point-to-point network, δ-RMT is impossible. It is straightforward that $P_1^{(+)} \cup P_2^{(+)} = P$ in a multicast graph is the same as $P_1 \cup P_2 = P$ in a point-to-point graph. Indeed, it can be seen as if P is split into two parts, and the adversary can choose to completely separate S and R on either part.

Now assume there exists a δ-RMT protocol Π that transmits a message $m \in \mathbb{M}$ from S to R. The strategy of the adversary is to choose an $e \in_R \{1, 2\}$ and control the set A_e. During the execution of the protocol Π, the adversary acts as follows:

- If $e = 1$, then the adversary completely separates S and R on $P_1^{(+)}$ to make the views of S and R completely different, and simulates the protocol on $P_1^{(+)}$ as S sent a message $m' \in \mathbb{M}$ to R.
- If $e = 2$, then the adversary completely separates S and R on $P \setminus P_1^{(+)}$ to make the views of S and R completely different. This is possible because $P_1^{(+)} \cup P_2^{(+)} = P \Rightarrow P \setminus P_1^{(+)} \subseteq P_2^{(+)}$. The adversary then simulates the protocol on $P \setminus P_1^{(+)}$ as S sent a message $m' \in \mathbb{M}$ to R.

Let m^R be the message that R recovers, since R does not know whether $e = 1$ or $e = 2$, similar to the proof in [FW00, Theorem 5.1], we have

$$\Pr[m^R = m | m' \neq m] \leq \Pr[m^R = m' | m' \neq m] \leq \Pr[\Pi \text{ fails } | m' \neq m].$$

This implies that the probability δ that the protocol Π fails is at least $\frac{1}{2}\Pr[m' \neq m] = \frac{1}{2}(1 - \frac{1}{|\mathbb{M}|})$. We have a contradiction. □

Lemma 5.2.2. *The condition of Theorem 5.2.1 is necessary.*

Proof. It is straightforward that in order to achieve δ-reliability, it is necessary to have $P_i \neq P$ for any $A_i \in \mathcal{A}$; i.e., $P \setminus P_i \neq \emptyset$.

Next we prove the necessity of the condition by contradiction. We assume that S and R are lowly $2\mathcal{A}$-separated (i.e., there exist two sets $A_1, A_2 \in \mathcal{A}$ as they are in Definition 5.1.6) and there exists a δ-RMT protocol Π that transmits a message $m \in \mathbb{M}$ from S to R. Without loss of generality, we let $P_1 \cap P_2 = \emptyset$. Now if $P_1^{(1)} = \emptyset$ and $P_2^{(1)} = \emptyset$, then we have $P_1^{(+)} = P_1$ and $P_2^{(+)} = P_2$, and hence $P_1^{(+)} \cup P_2^{(+)} = P$ (following Definition 5.1.6(a)), thus due to Lemma 5.2.1, δ-RMT is impossible in this case. In the rest of this proof we let $P_1^{(1)} \neq \emptyset$ and/or $P_2^{(1)} \neq \emptyset$.

During the execution of any protocol Π, we consider a node v. Throughout the protocol, v will multicast a tuple of elements, say $(v \sim \Pi)$. Note that some elements in $(v \sim \Pi)$ are not necessarily initiated by v, rather, they can just be initiated by other nodes and transmitted via v. If a part of the elements in $(v \sim \Pi)$ are multicast by all nodes (excluding S and R) on a path p (v is on p), then we use $(v \sim p)$ to denote the tuple of these elements. Note that some $(v \sim p)$ must exist, because otherwise any element transmitted during the protocol cannot be leant by both S and R. Now we let $(v \sim p)^S$ and $(v \sim p)^R$ be the views of S and R respectively on $(v \sim p)$.

The strategy of the adversary is to choose an $e \in_R \{1, 2\}$ and control the set A_e. Let $d \in \{1, 2\}$ such that $d \neq e$, then R should be able to recover the actual message from the elements received on P_d with probability $1 - \delta$. If, despite whether $e = 1$ or $e = 2$, $(v \sim p)^S \neq (v \sim p)^R$ for any v on any $p \in P_e$ (i.e., the views of S and R are completely different on P_e), then following the proof of Lemma 5.2.1, δ-RMT is impossible. Therefore, there must exist an $e \in \{1, 2\}$ such that $(v \sim p)^S = (v \sim p)^R$ is guaranteed for some v on some $p \in P_e$. We say that the tuple of elements $(v \sim p)$ where $p \in P_e$ such that $(v \sim p)^S = (v \sim p)^R$ is used to *support* the actual message. Following Lemma 5.1.2, the adversary can completely separate S and R on $P_e^{(+)}$ and cause $\forall(p \in P_e^{(+)}, v \text{ on } p) : (v \sim p)^S \neq (v \sim p)^R$. Following Lemma 5.1.3, for any path $p \in P_e^{(1)}$ (if $P_e^{(1)} \neq \emptyset$), $(v \sim p)^S = (v \sim p)^R$ can only be guaranteed if $v = A_e \sqcap p$. Therefore, there must exist an $e \in \{1, 2\}$ such that the actual message received on P_d is supported by some $((A_e \sqcap p) \sim p)$ where $p \in P_e^{(1)}$. Next, following Definition 5.1.6(b,c), for each path $p \in P_d^{(1)}$ (if $P_d^{(1)} \neq \emptyset$), we have case 1: $p \in P_e \cup P_e^{(*)}$, or case 2: $A_d \sqcap p$ has a neighbour in A_e. In case 1: $p \in P_e \cup P_e^{(*)}$, due to Lemma 5.1.1, there is no private transmission on path p whatsoever, so the adversary can learn $((A_d \sqcap p) \sim p)$. In case 2: $A_d \sqcap p$ has a neighbour in A_e, it is trivial that the adversary can learn $((A_d \sqcap p) \sim p)$.

To sum up, we can *conclude* that when the adversary chooses A_e to control, the actual message, which can be recovered from the elements received on P_d, should be supported by some $((A_e \sqcap p) \sim p)$ where $p \in P_e^{(1)}$ (if $P_e^{(1)} \neq \emptyset$), and the adversary can learn $((A_d \sqcap p) \sim p)$ for each $p \in P_d^{(1)}$ (if $P_d^{(1)} \neq \emptyset$). Note that $\{d, e\} = \{1, 2\}$.

Now during the execution of the protocol Π, the adversary corrupts P_e and causes $(v \sim p)^S \neq (v \sim p)^R$ for all nodes v on all paths $p \in P_e$ except for $p \in P_e^{(1)}$ and $v = A_e \sqcap p$. This is possible due to Lemma 5.1.2 and Lemma 5.1.3. As we *concluded* above, the adversary can always learn $((A_d \sqcap p) \sim p)$ for each $p \in P_d^{(1)}$. Thus on P_e, the adversary simulates the protocol as S sent a message $m' \in \mathbb{M}$, and m' can be supported by $((A_d \sqcap p) \sim p)$, where $p \in P_d^{(1)}$.

Therefore, at the end of the protocol Π, despite whether $e = 1$ or $e = 2$, the view of R always consists of the following:

- on P_1, a message is recovered which can be supported by $((A_2 \sqcap p) \sim p)$ for any $p \in P_2^{(1)}$ (if $P_2^{(1)} \neq \emptyset$), but may not be supported by any other elements received on P_2;
- on P_2, a different message is recovered which can be supported by $((A_1 \sqcap p) \sim p)$ for any $p \in P_1^{(1)}$ (if $P_1^{(1)} \neq \emptyset$), but may not be supported by any other elements received on P_1.

Thus as shown in Lemma 5.2.1, with probability $\delta \geq \frac{1}{2}(1 - \frac{1}{|\mathbb{M}|})$, R recovers the wrong message m'. This is a contradiction, which proves the necessity of the low $2\mathcal{A}$-connectivity. □

Next, to prove that the condition is sufficient, we give a δ-RMT protocol. Let $P = \{p_1, \ldots, p_n\}$, we first generalize some of Franklin and Wright's protocols in multicast graphs.

Full Distribution Protocol

1. For each $1 \leq j \leq n$, the nodes on path p_j execute an instance of the Single Path Distribution Protocol for each node v_i on p_j to distribute an element $a_{i,j}$. The nodes not on p_j do not multicast anything.
2. At the end of the protocol, on each path p_j ($1 \leq j \leq n$), S and R receive $a_{i,j}^S$ and $a_{i,j}^R$ respectively as the element initiated by node v_i on p_j. **End.**

Private Propagation Protocol

1. For each $1 \leq j \leq n$, the nodes on path p_j execute an instance of the Single Path Private Propagation Protocol from S to R in which S sends an element a_j^S, and the nodes not on p_j do not multicast anything.
2. At the end of the protocol, on each path p_j ($1 \leq j \leq n$), R receives a_j^R as the element that S initiated and propagated on p_j. **End.**

Now we give the following protocol which achieves δ-RMT for a message $m \in \mathbb{M}$ in a multicast graph $G(V, E)$.

Reliable Transmission Protocol

1. The nodes of V execute an instance of the Full Distribution Protocol in which for each $1 \leq j \leq n$, the elements that node v_i on path p_j initiates are $(a_{i,j}, b_{i,j}) \in_R \mathbb{F}^2$. Let $(a^S_{i,j}, b^S_{i,j})$ and $(a^R_{i,j}, b^R_{i,j})$ be what S and R receive respectively regarding $(a_{i,j}, b_{i,j})$.
2. The nodes of V execute an instance of the Private Propagation Protocol from S to R in which S sends the same vector on all paths in P:

$$(m, \langle \mathrm{auth}(m; a^S_{i,j}, b^S_{i,j}) \rangle)$$

   where $\langle \mathrm{auth}(m; a^S_{i,j}, b^S_{i,j}) \rangle$ is an ordered set of the encrypted m with *all* keys $(a^S_{i,j}, b^S_{i,j})$ that S receives in Step 1. At the end of the instance, R receives a vector $(m_k, \langle u_{i,j,k} \rangle)$ on each path $p_k \in P$.
3. Given the vector $(m_k, \langle u_{i,j,k} \rangle)$ that R receives on p_k, if $\exists (i, j) : u_{i,j,k} = \mathrm{auth}(m_k; a^R_{i,j}, b^R_{i,j})$, then we say that m_k is *qualified* on $(v_i \sim p_j)$. R finds an $A_f \in \mathcal{A}$ that satisfies the following three α-conditions:

   α-1 all vectors received on $P \setminus P_f$ are the same. We call this vector $(m_l, \langle u_{i,j,l} \rangle)$;

   α-2 $P^{(1)}_f = \emptyset$, or for each $p_j \in P^{(1)}_f$, we have that m_l is qualified on $((A_f \sqcap p_j) \sim p_j)$;

   α-3 $P \setminus (P_f \cup P^{(*)}_f) = \emptyset$, or for any vector $(m_k, \langle u_{i,j,k} \rangle)$ that R receives on path $p_k \in P_f$ such that $m_k \neq m_l$, we have that m_k is *not* qualified on any $(v_i \sim p_j)$ where $p_j \in P \setminus (P_f \cup P^{(*)}_f)$ and v_i does not have a neighbour in A_f.

   R then outputs the message m_l. **End.**

Lemma 5.2.3. *The Reliable Transmission Protocol is a δ-RMT protocol under the condition of Theorem 5.2.1.*

Proof. It is straightforward that if a corrupted m_k is qualified on some $(v_i \sim p_j)$ unknown to the adversary, then the Reliable Transmission Protocol fails. We use $\overline{RT}$ to denote the event when the above failure occurs and RT to denote the event otherwise. Let y be the number of nodes on the longest path (i.e., with the maximum number of nodes) between S and R, following the proof of Franklin and Wright [FW00, Theorem 3.4], the probability that the protocol fails is $\Pr[\overline{RT}] < \frac{yn^2}{|\mathbb{F}|}$. Thus this probability can be made negligible in security parameters (assuming $\mathbb{F}$ to be sufficiently large). Next in our proof, we assume that the above failure does not happen. That is, *we analyse the protocol in the event RT.*

The protocol achieves δ-reliability in Step 3. In the following, we first show that R can always find an $A_f \in \mathcal{A}$ that satisfies the three α-conditions, then we prove, by contradiction, that in the event RT, the message output by R is correct.

First, we show that there always exists an A_f that satisfies all three α-conditions, at least when the adversary chooses A_f to control so that P_f is *corrupted.* Since $P_f \neq P$ (following Lemma 5.1.5), we immediately have that condition α-1 is satisfied and m_l received on $P \setminus P_f$ is the actual message. If $P_f^{(1)} \neq \emptyset$, then as shown in the proof of Lemma 5.1.2, on each $p_j \in P_f^{(1)}$, S and R always have the same view on the key initiated by $A_f \sqcap p_j$. Thus it is clear that m_l is qualified on $((A_f \sqcap p_j) \sim p_j)$, and hence condition α-2 is satisfied. If $P \setminus (P_f \cup P_f^{(*)}) \neq \emptyset$, then the adversary cannot learn the key initiated by any node v_i which is on a path $p_j \in P \setminus (P_f \cup P_f^{(*)})$ if v_i does not have a neighbour in A_f. Thus without the above mentioned failure $\overline{RT}$, any faulty message $m_k \neq m_l$ cannot be qualified on such $(v_i \sim p_j)$, and hence condition α-3 is satisfied.

Next, using contradiction, we show that in the event RT, the message m_l that S outputs is the actual message. We assume that m_l is modified by the adversary who chooses a set $A_e \in \mathcal{A}$ to control, and all three α-conditions are satisfied. We now show that the three α-conditions imply the three properties of A_1, A_2 in Definition 5.1.6.

- From condition α-1, since all vectors received on $P \setminus P_f$ are modified, we have $P_e \cup P_f = P$ (i.e., corresponding to Definition 5.1.6(a)).

- Condition α-2 indicates that either $P_f^{(1)} = \emptyset$, or the adversary can learn the key initiated by node $A_f \sqcap p_j$ on any path $p_j \in P_f^{(1)}$ to make the faulty message m_l qualified on $((A_f \sqcap p_j) \sim p_j)$. Due to Lemma 5.1.4, this means that the adversary can separate S and R on p_j, completely eavesdrop on p_j or control a neighbour of $A_f \sqcap p_j$. Thus from condition α-2 we can conclude that $P_f^{(1)} = \emptyset$, or for each path $p_j \in P_f^{(1)}$, we have $p_j \in P_e \cup P_e^{(*)}$ or $A_f \sqcap p_j$ has a neighbour in A_e (i.e., corresponding to Definition 5.1.6(c)).

- Finally, since $P_e \neq P$ and $P_e \cup P_f = P$, there exists at least one path $p_k \in P_f$ such that the message m_k received on p_k is the actual message. Due to condition α-3, there are two cases:

case 1 $P \setminus (P_f \cup P_f^{(*)}) = \emptyset$, thus we have $P_e^{(1)} \subseteq P_f \cup P_f^{(*)} = P$.

case 2 The actual message m_k is *not* qualified on any $(v_i \sim p_j)$ where $p_j \in P \setminus (P_f \cup P_f^{(*)})$ and v_i does not have a neighbour in A_f. This implies that either $p_j \in P_e^{(+)}$, or $p_j \in P_e^{(1)}$ but any v_i on p_j that does not have a neighbour in A_f is not $A_e \sqcap p_j$ (because otherwise the actual message m_k should be qualified on $(v_i \sim p_j)$, due to the proof of Lemma 5.1.2). That is, if such $p_j \in P_e^{(1)}$ exists, then all the nodes on p_j that do not have a neighbour in A_f are not $A_e \sqcap p_j$. This implies that $A_e \sqcap p_j$ has a neighbour in A_f.

It is easy to conclude that in either case, $P_e^{(1)} = \emptyset$, or for each path $p_j \in P_e^{(1)}$, we have $p_j \in P_f \cup P_f^{(*)}$ or $A_e \sqcap p_j$ has a neighbour in A_f (i.e., corresponding to Definition 5.1.6(b)).

To sum up, under our assumption (i.e., the recovered message m_l is corrupted), A_e, A_f are as A_1, A_2 in Definition 5.1.6. This means S and R are lowly $2\mathcal{A}$-separated, which contradicts the condition of Theorem 5.2.1.

Therefore, at the end of the Reliable Transmission Protocol, R can recover $m_l = m$ with an negligible probability of failure (i.e., $\Pr[\overline{RT}] < \frac{yn^2}{|\mathbb{F}|}$). Thus the Reliable Transmission Protocol is a δ-RMT protocol. □

5.2.2 Perfectly Reliable Multicast

Here we show the necessary and sufficient condition for 0-RMT in multicast graphs.

Theorem 5.2.2. *Given a graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, the necessary and sufficient condition for 0-RMT from S to R is that S and R are $2\mathcal{A}$-connected in G.*

Proof. The proof of necessity straightforwardly follows Franklin and Wright's proof of [FW00, Theorem 3.6]. Indeed, assume that there exist two sets $A_1, A_2 \in \mathcal{A}$ such that $P_1 \cup P_2 = P$, the adversary who chooses A_e ($e \in_R \{1, 2\}$) to control will simulate the protocol on P_e to transmit a faulty message. During the transmission, the adversary will guess the information it cannot learn to support its faulty message. Thus at the end of *any* protocol, R has to distinguish two different messages received respectively from P_1 and P_2. With some non-zero probability, the guess of the adversary during the execution of the protocol is valid, and R, in this case (or, as denoted in the proof of Lemma 5.2.2, the event $\overline{RT}$), may mistakenly recover the wrong message. Thus there does not exist a protocol that is 0-reliable.

The sufficiency of the condition can be proven with a 0-RMT protocol that combines our Private Propagation Protocol with Desmedt et al.'s 0-RMT protocol [DWB05] (see also Appendix A.4). The protocol is briefly described as follows. The nodes of V execute an instance of the Private Propagation Protocol from S to R in which S sends the message m on all paths in P. R finds an $A_f \in \mathcal{A}$ such that all messages received on $P \setminus P_f$ are equal. R sets m' to be this message. It is trivial that $m' = m$, and hence this protocol achieves 0-reliability. □

Evidently the necessary and sufficient condition for 0-RMT in the multicast setting is the same as that in the point-to-point setting. Same result has been shown by Franklin and Wright [FW98] in the threshold model (see Table 2.1).

5.3 Secure Multicast Communication

In this section we take the problem of achieving privacy into consideration. Therefore, the SMT protocols should guarantee some level of privacy while enabling RMT. We study almost perfect security in Section 5.3.1. In this case, δ-reliability is required, and in addition, ϵ-privacy or perfect privacy should also be guaranteed. Thus we discuss both (ϵ, δ)-SMT and $(0, \delta)$-SMT. In Section 5.3.2, we study $(0, 0)$-SMT that enables perfect security.

5.3.1 Almost Perfectly Secure Multicast

First, we give the necessary and sufficient condition for (ϵ, δ)-SMT in multicast graphs. Unlike the neighbour-disjoint setting in [FW98] where the conditions for both δ-RMT and (ϵ, δ)-SMT are the same (i.e., $n > t$), in multicast graphs, (ϵ, δ)-SMT requires stronger connectivity than that for δ-RMT.

Theorem 5.3.1. *Given a graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, the necessary and sufficient condition for (ϵ, δ)-SMT from S to R is that S and R are highly $\mathcal{A}$-connected and lowly $2\mathcal{A}$-connected.*

Proof. We first prove the necessity of the condition. It is straightforward that the high $\mathcal{A}$-connectivity, i.e., $P_i \cup P_i^{(*)} \neq P$, is necessary for achieving ϵ-privacy, because otherwise there is no private transmission between S and R on any path in P whatsoever. Moreover, as proven by Lemma 5.2.2, the low $2\mathcal{A}$-connectivity is necessary for achieving δ-reliability. Thus the condition is necessary for (ϵ, δ)-SMT.

Next, we show that the condition is sufficient. Let $P = \{p_1, \ldots, p_n\}$, we give the following protocol for S to send a message $m \in \mathbb{M}$ to R.

Private Transmission Protocol

1. The nodes of V execute an instance of the Private Propagation Protocol from S to R in which for each $1 \leq j \leq n$, S sends a pair $(a_j^S, b_j^S) \in_R \mathbb{F}$ on path $p_j \in P$. At the end of the instance, R receives a pair (a_j^R, b_j^R) on each path $p_j \in P$.
2. R chooses an element $r^R \in_R \mathbb{F}$ and for each $1 \leq j \leq n$, computes $s_j^R = \text{auth}(r^R; a_j^R, b_j^R)$. The nodes of V execute an instance of the Reliable Transmission Protocol from R to S in which R sends a vector $(r^R, s_1^R, \ldots, s_n^R)$. At the end of the instance, S outputs a vector $(r^S, s_1^S, \ldots, s_n^S)$.
3. S computes an index set $I = \{j | s_j^S = \text{auth}(r^S; a_j^S, b_j^S)\}$ and an encryption key $key = \sum_{j \in I} a_j^S$, and then encrypts the message $c = m + key$.

The nodes of V execute an instance of the Reliable Transmission Protocol from S to R in which S sends a vector (I, c). At the end of the instance, R outputs a vector (I', c').

4. R computes a decryption key $key' = \sum_{j \in I'} a_j^R$ and decrypts the message $m' = c' - key'$. **End.**

First we show that this protocol is ϵ-private under the condition. Suppose that the adversary chooses a set A_e to control. Since $P_e \cup P_e^{(*)} \neq P$ (high $\mathcal{A}$-connectivity), there exists a path $p_d \in P \setminus (P_e \cup P_e^{(*)})$ on which the adversary cannot completely eavesdrop. As shown in the proof of Lemma 5.1.1, the adversary cannot learn (a_d^S, b_d^S) in Step 1. Because $p_d \notin P_e$, we have $(a_d^R, b_d^R) = (a_d^S, b_d^S)$. Let RT denote the event that the instance of the Reliable Transmission Protocol in Step 2 succeeds and $\overline{RT}$ denote the event otherwise. In the event RT, $r^S = r^R$ and for each $1 \leq j \leq n$, we have $s_j^S = s_j^R$. This implies that $d \in I$. The adversary who cannot learn a_d^S by eavesdropping or by decoding s_d^R (due to Theorem 2.8.1) will not be able to compute key to decrypt m. That is, for any two messages $m_0, m_1 \in \mathbb{M}$ and any coin flips r, we have the following:

$$\textstyle\sum_c |\Pr[adv(m_0, r) = c|RT] - \Pr[adv(m_1, r) = c|RT]| = 0 \qquad (5.1)$$

$$\textstyle\sum_c |\Pr[adv(m_0, r) = c|\overline{RT}] - \Pr[adv(m_1, r) = c|\overline{RT}]| \leq |+1| + |-1| = 2 \qquad (5.2)$$

Let $\Pr[\overline{RT}] = \epsilon$, that is, ϵ is the probability with which the instance of the Reliable Transmission Protocol in Step 2 fails. This can be made negligible as discussed in the proof of Lemma 5.2.3. By combining Eq. 5.1 and Eq. 5.2, we have the following:

$$\textstyle\sum_c |\Pr[adv(m_0, r) = c] - \Pr[adv(m_1, r) = c]| \leq 0 \cdot \Pr[RT] + 2 \cdot \Pr[\overline{RT}] = 2\epsilon.$$

Thus the Private Transmission Protocol is ϵ-private (see notation in Section 2.4).

Next we show that the protocol is δ-reliable. Let δ_1 be the probability that the instance of the Reliable Transmission Protocol in Step 2 fails and δ_2 be the probability that the instance in Step 3 fails. As shown in the proof of Lemma 5.2.3, δ_1 and δ_2 can be made negligible in security parameters. Let δ_3 be the probability that both the above instances succeed, but R outputs $m' \neq m$. This can only happen if there exists at least one $j \in I$ such that $a_j^S \neq a_j^R$. Since both reliable protocols succeed, the fact $j \in I$ implies $\text{auth}(r^R; a_j^S, b_j^S) = \text{auth}(r^R; a_j^R, b_j^R)$. That is,

$$a_j^S + b_j^S r^R = a_j^R + b_j^R r^R \Rightarrow r^R = \frac{a_j^R - a_j^S}{b_j^S - b_j^R} \in \mathbb{F}, \qquad (5.3)$$

where $b_j^S \neq b_j^R$.[3] Since r^R is chosen with respect to the uniform distribution, if the adversary modifies (a_j^S, b_j^S) to (a_j^R, b_j^R) on path p_j in Step 1, then the probability that

[3] If $b_j^S = b_j^R$, then $\text{auth}(r^R; a_j^S, b_j^S) = \text{auth}(r^R; a_j^R, b_j^R)$ implies $a_j^S = a_j^R$. Since $a_j^S \neq a_j^R$, we always have $b_j^S \neq b_j^R$.

Eq. 5.3 is fulfilled is $\frac{1}{|\mathbb{F}|}$. Since the adversary can corrupt $|P_e|$ paths, it is straightforward that $\delta_3 = \frac{|P_e|}{|\mathbb{F}|} < \frac{n}{|\mathbb{F}|}$, which is much smaller than δ_1 and δ_2. Thus the final probability that the protocol fails to be reliable is

$$\delta = \delta_1 + (1 - \delta_1)\delta_2 + (1 - (\delta_1 + (1 - \delta_1)\delta_2))\delta_3 < \delta_1 + \delta_2 + \delta_3.$$

Therefore, the Private Transmission Protocol is an (ϵ, δ)-SMT protocol. This implies that the condition is sufficient, which concludes our proof. □

Note that compared to δ-RMT, (ϵ, δ)-SMT requires an extra condition: the high $\mathcal{A}$-connectivity. This is because, following our discussion in the proof of Lemma 5.1.1, the condition $P_i \cup P_i^{(*)} \neq P$ corresponds to the weak t_{hyper}-connectivity in hypergraphs, which is necessary for private communication as proven by Franklin and Yung [FY95]. Therefore, we can see the condition of Theorem 5.3.1 as that it consists of two parts, with the high $\mathcal{A}$-connectivity enabling private communication and the low $2\mathcal{A}$-connectivity enabling δ-reliable communication. These two types of connectivities are *independent*. Indeed, with some examples in Section 5.4, we can show that they do not imply each other.

From Corollary 4.2.1, we know that reducing the requirement for privacy does not weaken the minimal connectivity. Thus in the following theorem, we show that the condition for (ϵ, δ)-SMT is also necessary and sufficient for $(0, \delta)$-SMT.

Theorem 5.3.2. *Given a graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, the necessary and sufficient condition for $(0, \delta)$-SMT from S to R is that S and R are highly $\mathcal{A}$-connected and lowly $2\mathcal{A}$-connected.*

Proof. It is trivial that the condition is necessary following the proof of Theorem 5.3.1. Next we show that the condition is sufficient by slightly amending the Private Transmission Protocol to the following protocol which achieves perfect privacy.

Perfectly Private Transmission Protocol

1. Same as Step 1 in the Private Transmission Protocol.
2. R chooses an element $r^R \in_R \mathbb{F}$ and for each $1 \leq j \leq n$, computes $s_j^R = \text{auth}(r^R; a_j^R, b_j^R)$. The nodes of V executes an instance of the Reliable Transmission Protocol from R to S in which R sends a vector $(r^R, s_1^R, \ldots, s_n^R)$. At the end of the instance, S distinguishes the following two cases:

Case 1 There exist two sets $A_{f_1}, A_{f_2} \in \mathcal{A}$ that satisfy all three α-conditions of the Reliable Transmission Protocol, and the vectors u_1 and u_2 (both regarding the vector $(r^R, s_1^R, \ldots, s_n^R)$) that S receives respectively on $P \setminus P_{f_1}$ and $P \setminus P_{f_2}$ are different, then S terminates the protocol.

Case 2 Otherwise, S outputs a vector $(r^S, s_1^S, \dots, s_n^S)$, and goes to Step 3.

3. Same as Step 3 in the Private Transmission Protocol.
4. Same as Step 4 in the Private Transmission Protocol. **End.**

Now we show that this protocol is 0-private. Following the proof of Theorem 5.3.1, the privacy of the message transmission can only be breached in the event $\overline{RT}$. It is clear that the instance of the Reliable Transmission Protocol in Step 2 allows S to distinguish between RT and $\overline{RT}$. As shown in the proof of Lemma 5.2.3, in the event RT, only the correct vector can be output after the Reliable Transmission Protocol. This means that if two different vectors can be output, then the event $\overline{RT}$ occurs. Thus in Step 2, Case 1 indicates $\overline{RT}$ and Case 2 indicates RT. In the event $\overline{RT}$, S terminates the protocol so the adversary learns nothing about the message. Corresponding to Eq. 5.2 in the previous proof, we have the following:

$$\textstyle\sum_c |\Pr[adv(m_0, r) = c|\overline{RT}] - \Pr[adv(m_1, r) = c|\overline{RT}]| = 0. \tag{5.4}$$

Using Eq. 5.1 and Eq. 5.4, we have the following:

$$\textstyle\sum_c |\Pr[adv(m_0, r) = c] - \Pr[adv(m_1, r) = c]| = 0 \cdot \Pr[RT] + 0 \cdot \Pr[\overline{RT}] = 0.$$

Thus the protocol achieves 0-privacy.

Next, using a similar proof to that of Theorem 5.3.1, we can prove that the Perfectly Private Transmission Protocol is also δ-reliable. This concludes the proof. □

5.3.2 Perfectly Secure Multicast

In [DDWY93], Dolev et al. showed that if σ is the maximum number of channels that a *listening* (passive) adversary can control and τ is the maximum number of channels that a *disrupting* (active) adversary can control, then there must be at least $n \geq \max\{\sigma + \tau + 1, 2\tau + 1\}$ paths between S and R in an undirected point-to-point graph for PSMT (i.e., $(0,0)$-SMT) to be possible. This setting can be generalized in our model as follows: given an adversary structure $\mathcal{A} = \{A_1, \dots, A_z\}$, then $\{P_1 \cup P_1^{(*)}, \dots, P_z \cup P_z^{(*)}\}$ consists of the subsets of paths a listening adversary can control and $\{P_1, \dots, P_z\}$ consists of the subsets of paths a disrupting adversary can control. Thus we give the following theorem for $(0,0)$-SMT in multicast graphs.

Theorem 5.3.3. *Given a graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$, the necessary and sufficient condition for $(0,0)$-SMT from S to R is that S and R are highly $2\mathcal{A}$-connected in G.*[4]

Proof. Using a similar proof to that of [DDWY93, Theorem 5.2], we can prove that the condition is necessary. To prove that the condition is sufficient, we can easily combine

[4] The high $2\mathcal{A}$-connectivity means $(P_i \cup P_i^{(*)}) \cup P_j \neq P$ for any $A_i, A_j \in \mathcal{A}$. See Definition 5.1.5.

the Private Propagation Protocol with any of the PSMT protocols in Section 4.4 to give a (0,0)-SMT protocol under this condition. In this case, the linear code should be constructed with respect to the critical-path structure $\mathcal{P}^* = \{P_1 \cup P_1^{(*)}, \dots, P_z \cup P_z^{(*)}\}$, and the errors are caused by a set in the critical path-structure $\mathcal{P} = \{P_1, \dots, P_z\}$. □

Again from the result shown in Corollary 4.2.1, we can determine the minimal connectivity for $(\epsilon, 0)$-SMT.

Corollary 5.3.1. *Given a graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A} = \{A_1, \dots, A_z\}$ be an adversary structure on $V \setminus \{S, R\}$, the necessary and sufficient condition for $(\epsilon, 0)$-SMT from S to R is that S and R are highly $2\mathcal{A}$-connected in G.*

5.4 Multicast in the Threshold Model

In this section we use our results in the general adversary model to find the necessary and sufficient conditions for SMT in the threshold model. Unlike those in [FW98, FW00], our results are obtained in general multicast graphs without the requirement of node-disjoint and neighbour-disjoint paths.

First, because a threshold is a special case of an adversary structure, we re-define the threshold model in the adversary structure context.

Definition 5.4.1. *Given a graph $G(V, E)$, a threshold t is an adversary structure $\mathcal{A}^T \subseteq 2^{V \setminus \{S,R\}}$ such that $\forall (A \subseteq V \setminus \{S, R\}, |A| \leq t) : A \in \mathcal{A}^T$.*

Using this new definition of the threshold, we define two special types of connectivities in the threshold model as follows.

Definition 5.4.2. *Given a graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}^T$ be a threshold t on $V \setminus \{S, R\}$,*

- *we say S and R are $t_{\zeta\text{-}private}$-*connected *if they are highly $\mathcal{A}^T$-connected in G;*
- *we say S and R are $t_{\zeta\text{-}reliable}$-*connected *if they are lowly $2\mathcal{A}^T$-connected in G.*

Next, we show that our results correspond to Franklin and Wright's results in [FW98] (see Table 2.2) if the multicast graph consists of *only* node-disjoint and neighbour-disjoint paths.

Lemma 5.4.1. *In a multicast graph that consists of n node-disjoint and neighbour-disjoint paths, $n > t$ implies the conditions of Theorem 5.2.1, Theorem 5.3.1 and Theorem 5.3.2, and $n > 2t$ implies the conditions of Theorem 5.2.2 and Theorem 5.3.3.*

Proof. For each $A_i \in \mathcal{A}^T$, we let $P_i^{(0)} \subseteq P \setminus P_i$ be the set of all paths on each of which there is at least one node that have a neighbour in A_i (thus $P_i^{(*)} \subseteq P_i^{(0)}$). In such a special multicast graph, each node can only be on or be a neighbour of at most one path. Thus we have $|P_i \cup P_i^{(0)}| \leq t$. Also, because there are at least two nodes on each path in $P_i^{(+)}$, we have $2|P_i^{(+)}| + |P_i^{(1)}| + |P_i^{(0)}| \leq t$.

First, it is straightforward that $n > 2t$ implies the conditions of Theorem 5.2.2 and Theorem 5.3.3. Since $n > 2t$ means that for any $A_i, A_j \in \mathcal{A}^T$, we have $(P_i \cup P_i^{(0)}) \cup (P_j \cup P_j^{(0)}) \neq P$, the condition of Theorem 5.2.2, $P_i \cup P_j \neq P$, and the condition of Theorem 5.3.3, $(P_i \cup P_i^{(*)}) \cup P_j \neq P$, are clearly satisfied.

Next, $n > t$ means that for any $A_i \in \mathcal{A}^T$, we have $P_i \cup P_i^{(0)} \neq P$. Obviously this implies the $t_{\zeta\text{-}private}$-connectivity which means $P_i \cup P_i^{(*)} \neq P$. Thus to show that $n > t$ implies the conditions of Theorem 5.2.1, Theorem 5.3.1 and Theorem 5.3.2, we only need to prove that $n > t$ implies the $t_{\zeta\text{-}reliable}$-connectivity.

Finally, we prove this implication by contradiction. Assume that S and R are not $t_{\zeta\text{-}reliable}$-connected; i.e., they are not lowly $2\mathcal{A}^T$-connected. That is, there exist two sets $A_1, A_2 \in \mathcal{A}^T$ such that $P_1 \cup P_2 = P$ (Definition 5.1.6(a)). From Definition 5.1.6(b), we have $P_1^{(1)} \subseteq P_2 \cup P_2^{(0)}$, and hence $P_2^{(+)} \cup P_2^{(1)} \cup P_2^{(0)} \cup P_1^{(+)} = P$ because

$$P_1^{(+)} \cup P_1^{(1)} \cup P_2^{(+)} \cup P_2^{(1)} = P \text{ and } P_1^{(1)} \subseteq P_2^{(+)} \cup P_2^{(1)} \cup P_2^{(0)}.$$

Similarly, from Definition 5.1.6(c), we have $P_1^{(+)} \cup P_1^{(1)} \cup P_1^{(0)} \cup P_2^{(+)} = P$. Therefore, we have $|P_1^{(+)}| + |P_1^{(1)}| + |P_1^{(0)}| + |P_2^{(+)}| \geq n$ and $|P_2^{(+)}| + |P_2^{(1)}| + |P_2^{(0)}| + |P_1^{(+)}| \geq n$, and hence

$$(2|P_1^{(+)}| + |P_1^{(1)}| + |P_1^{(0)}|) + (2|P_2^{(+)}| + |P_2^{(1)}| + |P_2^{(0)}|) \geq 2n. \tag{5.5}$$

As we showed at the beginning of this proof, for each $i \in \{1, 2\}$, we have $2|P_i^{(+)}| + |P_i^{(1)}| + |P_i^{(0)}| \leq t$. Thus Eq. 5.5 implies $t + t \geq 2n$, and hence $n \leq t$, which is a contradiction. This proves that $n > t$ implies the $t_{\zeta\text{-}reliable}$-connectivity. □

Next we discuss the connectivities in the general multicast graph setting with some results in the previous studies. First we remark the relations of the t-connectivity (see Definition 2.2.1), the $t_{neighbour}$-connectivity (see Definition 2.2.2), the weak (n, t)-connectivity (see Definition 2.2.3) and the weak t_{hyper}-connectivity (see Definition 2.2.4). In [DW02], Desmedt and Wang [DW02] showed that the following implications are strict (see Section 2.9.2):

$$\begin{aligned}\text{weak } (n,t)\text{-connectivity } &\Rightarrow t_{neighbour}\text{-connectivity}\\ &\Rightarrow \text{weak } t_{hyper}\text{-connectivity } \Rightarrow t\text{-connectivity.}\end{aligned}$$

As shown in the proof of Lemma 5.1.1, Franklin and Yung's weak t_{hyper}-connectivity in a hypergraph H_G is essentially our $t_{\zeta\text{-}private}$-connectivity in a multicast graph G. In [WD99], Wang and Desmedt claimed that their weak (n, t)-connectivity is sufficient for $(0, \delta)$-SMT. Since weak (n, t)-connectivity $\Rightarrow$ $t_{\zeta\text{-}private}$-connectivity, it is clear that

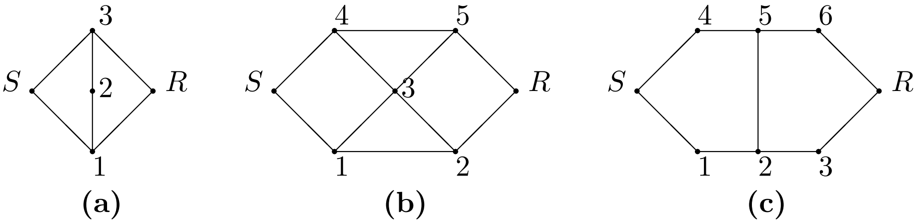


Figure 5.2: Private and reliable connectivity (duplicate of Figure 2.2).

0-privacy can be achieved. However, δ-reliability is only achievable under their condition if weak (n, t)-connectivity $\Rightarrow$ $t_{\zeta\text{-}reliable}$-connectivity. In [WD99], there is not a proper proof showing this implication. Thus their result (weak (n, t)-connectivity being the upper bound for $(0, \delta)$-SMT) is only a conjecture. We leave this as an open problem.

As discussed in Section 2.9.2, Desmedt and Wang [DW02, WD08] gave some results regarding SMT in multicast graphs with examples. Here, using Figure 5.2, we explain their examples and prove their conjectures as follows.

- **Result:** The weak (n, t)-connectivity is not necessary for $(0, \delta)$-SMT in multicast graphs. E.g., in Figure 5.2(a), S and R are not weakly $(2, 1)$-connected, but $(0, \delta)$-SMT is possible.[5]

  Our explanation: This is because S and R are obviously $1_{\zeta\text{-}private}$-connected and $1_{\zeta\text{-}reliable}$-connected in Figure 5.2(a).

- **Result:** The weak t_{hyper}-connectivity is not necessary for δ-RMT. E.g., in Figure 5.2(b), S and R are not weakly 1_{hyper}-connected, but δ-RMT is possible.

  Our explanation: This is because S and R are $1_{\zeta\text{-}reliable}$-connected but not $1_{\zeta\text{-}private}$-connected in Figure 5.2(b).

- **Conjecture:** The weak t_{hyper}-connectivity is not sufficient for $(0, \delta)$-SMT. E.g., in Figure 5.2(c), S and R are weakly 1_{hyper}-connected, but Desmedt and Wang conjectured that there is no $(0, \delta)$-SMT against a 1-bounded adversary.

  Our proof: Indeed, not only is $(0, \delta)$-SMT impossible, but δ-RMT is also impossible, because S and R are not $1_{\zeta\text{-}reliable}$-connected in Figure 5.2(c).

Note that the examples of Figure 5.2(b) and Figure 5.2(c) also show that the $t_{\zeta\text{-}private}$-connectivity and the $t_{\zeta\text{-}reliable}$-connectivity *do not imply each other*, because in Figure 5.2(b), S and R are $1_{\zeta\text{-}reliable}$-connected but not $1_{\zeta\text{-}private}$-connected, and in Figure 5.2(c), they are $1_{\zeta\text{-}private}$-connected but not $1_{\zeta\text{-}reliable}$-connected. These examples also show that the high $\mathcal{A}$-connectivity and the low $2\mathcal{A}$-connectivity are independent.

Finally in this section, we present the following corollary as the results for SMT in multicast graphs in the threshold model.

[5] We observe that their $(0, \delta)$-SMT protocol in [WD08] is actually ϵ-private ($\epsilon > 0$), but it is easy to fix the protocol.

Corollary 5.4.1. *Given a graph $G(V, E)$ where $S, R \in V$ and an adversary who can control up to t nodes in $V \setminus \{S, R\}$,*

- *δ-RMT is possible if and only if S and R are $t_{\zeta\textit{-reliable}}$-connected in G.*
- *0-RMT is possible if and only if S and R are $2t$-connected in G.*
- *(ϵ, δ)-SMT or $(0, \delta)$-SMT is possible if and only if S and R are $t_{\zeta\textit{-private}}$-connected and $t_{\zeta\textit{-reliable}}$-connected in G.*
- *$(\epsilon, 0)$-SMT or $(0, 0)$-SMT is possible if and only if S and R are $(t_{\zeta\textit{-private}} + t)$-connected in G. The $(t_{\zeta\textit{-private}} + t)$-connectivity means that S and R are highly $2\mathcal{A}^T$-connected.*

5.5 Brief Conclusion of Chapter 5

In this chapter, we found the necessary and sufficient conditions for SMT in multicast networks. As stated in Section 2.9.2, before this chapter, if we draw a similar table as Table 2.1 for SMT in general multicast graphs (without the requirement of neighbour-disjoint paths), then all the result fields will be N/A. With the minimal connectivities determined in this chapter, we can finally present such a table as Table 5.1, which summarizes our results for SMT in multicast networks. These results are presented using the Extended Characterization, which is presented based on our observation on the eavesdropping and separating activities of the adversary on a single path.

Multicast Graphs			
SMT	RMT	δ-RMT	$t_{\zeta\textit{-reliable}}$-conn. (Corollary 5.4.1)
			high $\mathcal{A}$-conn. (Theorem 5.2.1)
		0-RMT	$2t$-conn. (Corollary 5.4.1)
			$2\mathcal{A}$-conn. (Theorem 5.2.2)
	APSMT	(ϵ, δ)-SMT	$t_{\zeta\textit{-private}}$-conn. and $t_{\zeta\textit{-reliable}}$-conn. (Corollary 5.4.1)
			high $\mathcal{A}$-conn. and low $2\mathcal{A}$-conn. (Theorem 5.3.1)
		$(\epsilon, 0)$-SMT	$(t_{\zeta\textit{-private}} + t)$-conn. (Corollary 5.4.1)
			high $2\mathcal{A}$-conn. (Corollary 5.3.1)
		$(0, \delta)$-SMT	$t_{\zeta\textit{-private}}$-conn. and $t_{\zeta\textit{-reliable}}$-conn. (Corollary 5.4.1)
			high $\mathcal{A}$-conn. and low $2\mathcal{A}$-conn. (Theorem 5.3.2)
	PSMT	$(0, 0)$-SMT	$(t_{\zeta\textit{-private}} + t)$-conn. (Corollary 5.4.1)
			high $2\mathcal{A}$-conn. (Theorem 5.3.3)

* “conn.” is short for “connectivity”.

** For each security level (e.g., δ-RMT), the results are presented in two rows: the upper row indicates the result in the threshold model and the lower row indicates the result in the general adversary model.

Table 5.1: Network connectivities for SMT in multicast networks.

In the following concluding chapter, we summarize the results of this thesis, and propose future work.

Chapter 6

Conclusion and Future Work

We have considered the problem of secure message transmission (SMT) in the general adversary model. We began with introducing our ideas and observations that would help us determine minimal connectivities and design efficient protocols for SMT. Then we studied SMT in two different kinds of networks: point-to-point and multicast.

In this chapter, we summarize the results of this thesis in Section 6.1 and propose possible future research directions in Section 6.2.

6.1 Summary of Results

First of all, using exhaustive search, we proved that an almost perfect threshold $(t+1, n, \kappa)$-SSS can detect, and simultaneously correct, exactly the same numbers of errors as a perfect $(t+1, n, 0)$-SSS can. That is, a $(t+1, n, \kappa)$-SSS can detect $n-t-1$ errors and correct $\lfloor \frac{n-t-1}{2} \rfloor$ errors. This result was later used to prove that the minimal connectivity for almost perfect ϵ-private message transmission is the same as that for perfect 0-private transmission, when the same level of reliability is required.

Next, we studied generalized LSSSs and regarded general access structures as linear codes. Some properties of our linear code have been introduced, including the information of a codeword, the decoding condition and the error-correcting capability. In particular, we generalized the idea of pseudo-basis and pseudo-dimension using this new linear code. We defined the size and weight of an adversary structure, which are used to bound the pseudo-dimension and later used to determine the communication complexity (CC) of our protocols. The possible existence of the invalid error vectors has also been discussed as a crucial part of our result.

We formally defined the critical paths in a general graph, and observed their properties. From our observation, we concluded that if the SMT protocols are executed via the critical paths, then it is impossible to determine its CC in the size of the network, because the number of paths varies remarkably in different graphs with the same connectivity. Thus we defined a new adversary structure over critical paths, and used this

Undirected Graphs			
SMT	RMT	δ-RMT	$n \geq 2t + 1$ [FW98]
			$2\mathcal{A}$-conn. (Theorem 4.2.1)
		0-RMT	$n \geq 2t + 1$ [DDWY93]
			$2\mathcal{A}$-conn. [KGSR02]
	APSMT	(ϵ, δ)-SMT	$n \geq 2t + 1$ (Corollary 4.2.2)
			$2\mathcal{A}$-conn. (Corollary 4.2.2)
		$(\epsilon, 0)$-SMT	$n \geq 2t + 1$ (Corollary 4.2.2)
			$2\mathcal{A}$-conn. (Corollary 4.2.2)
		$(0, \delta)$-SMT	$n \geq 2t + 1$ [FW98]
			$2\mathcal{A}$-conn. (Theorem 4.2.1)
	PSMT	$(0, 0)$-SMT	$n \geq 2t + 1$ [DDWY93]
			$2\mathcal{A}$-conn. [KGSR02]

Directed Graphs			
SMT	RMT	δ-RMT	$n \geq \max\{2t + 1 - u, t + 1\}$ [DW02]
			strong $2\mathcal{A}$-directed-conn. (Theorem 4.2.2)
		0-RMT	$n \geq 2t + 1$ [DDWY93]
			$2\mathcal{A}$-conn. on forward paths [DWB05]
	APSMT	(ϵ, δ)-SMT	$n \geq \max\{2t + 1 - u, t + 1\}$ (Corollary 4.2.3)
			strong $2\mathcal{A}$-directed-conn. (Corollary 4.2.3)
		$(\epsilon, 0)$-SMT	$n \geq \max\{3t + 1 - 2u, 2t + 1\}$ (Corollary 4.2.3)
			strong $3\mathcal{A}$-directed-conn. (Corollary 4.2.3)
		$(0, \delta)$-SMT	$n \geq \max\{2t + 1 - u, t + 1\}$ [DW02]
			strong $2\mathcal{A}$-directed-conn. (Theorem 4.2.2)
	PSMT	$(0, 0)$-SMT	$n \geq \max\{3t + 1 - 2u, 2t + 1\}$ [DW02]
			strong $3\mathcal{A}$-directed-conn. [PSC+07]

Multicast Graphs			
SMT	RMT	δ-RMT	$t_{\zeta\text{-}reliable}$-conn. (Corollary 5.4.1)
			high $\mathcal{A}$-conn. (Theorem 5.2.1)
		0-RMT	$2t$-conn. (Corollary 5.4.1)
			$2\mathcal{A}$-conn. (Theorem 5.2.2)
	APSMT	(ϵ, δ)-SMT	$t_{\zeta\text{-}private}$-conn. and $t_{\zeta\text{-}reliable}$-conn. (Corollary 5.4.1)
			high $\mathcal{A}$-conn. and low $2\mathcal{A}$-conn. (Theorem 5.3.1)
		$(\epsilon, 0)$-SMT	$(t_{\zeta\text{-}private} + t)$-conn. (Corollary 5.4.1)
			high $2\mathcal{A}$-conn. (Corollary 5.3.1)
		$(0, \delta)$-SMT	$t_{\zeta\text{-}private}$-conn. and $t_{\zeta\text{-}reliable}$-conn. (Corollary 5.4.1)
			high $\mathcal{A}$-conn. and low $2\mathcal{A}$-conn. (Theorem 5.3.2)
	PSMT	$(0, 0)$-SMT	$(t_{\zeta\text{-}private} + t)$-conn. (Corollary 5.4.1)
			high $2\mathcal{A}$-conn. (Theorem 5.3.3)

[*] "conn." is short for "connectivity".

[**] For each security level (e.g., δ-RMT), the results are presented in two rows: the upper row indicates the result in the threshold model and the lower row indicates the result in the general adversary model.

Table 6.1: Complete network connectivities for SMT.

idea to design SMT protocols.

The results regarding SMT on point-to-point networks consist of the following: a Guessing Attack on some previous (directed) protocols, the necessary and sufficient conditions for APSMT and the constructions of some efficient PSMT protocols. First, the Guessing Attack can be performed in a directed graph against a PSMT protocol. By replacing the feedbacks from the receiver with some guessed values, the adversary is able to learn the messages with better probability, and hence breach perfect privacy of the protocol. Secondly, the necessary and sufficient conditions for APSMT have been obtained by generalizing some previous results and applying our above mentioned result on the error-correcting capability of a $(t+1, n, \kappa)$-SSS. These findings completed the research of determining the minimal connectivities in the point-to-point model. Finally, a number of efficient PSMT protocols have been constructed using the newly proposed generalized linear code. By comparing our results with the previous protocols in terms of CC and RC (see Table 4.1), the significance of our design is obvious.

The results regarding SMT on multicast networks consist of the following: an Extended Characterization of a multicast graph based on the eavesdropping and separating activities, the necessary and sufficient conditions for all levels of SMT in the general adversary model and the corresponding conditions in the threshold model. First, our observation on the eavesdropping and separating activities of the adversary allowed us to gain an insight on multicast communication, and hence helped us re-characterize the multicast graph and define respective network connectivities. Next, the Extended Characterization was later used to find the necessary and sufficient conditions for reliable and secure communication, and protocols were designed to enable such communication. Finally, we applied our findings in the general adversary model to determine the minimal network connectivities in the threshold model. These results explain all the examples shown in [DW02], prove the conjecture given in [DW02], and completely solve the open problem raised in [FW98].

Therefore, we have finally determined the minimal network connectivities for all levels of SMT in all kinds of point-to-point (undirected and directed) graphs and multicast graphs, regarding different adversary models. The complete results on network connectivities are shown in Table 6.1.

6.2 Future Work

In this section, we discuss the limitations of this thesis, and propose possible future research directions.

Evidently, there are still many unknown properties of the linear code described in Section 3.2. One of the most obvious unknown properties is the tight upper bound on the size of the codeword (i.e., h). This is a difficult problem which has been open for decades. Furthermore, we considered that the pseudo-dimension of our scheme is at most $wt^{\mathcal{A}} = O(h)$, because there are at most $wt^{\mathcal{A}}$ non-zero entries in each error vector. Now with the existence of non-zero invalid error vectors, the dimension of the vector space of the *valid* error vectors may be smaller than we expected. Thus another interesting problem is whether it is possible to have a pseudo-dimension smaller than $O(h)$ in the presence of non-zero invalid error vectors.

Next, in point-to-point networks, the CC of our 2-Round Undirected Protocol and 3-Round Directed Protocols for multiple message transmission is $O(hn\ell\rho)$ (see Table 4.1). In [SNR04, ACdH06, KS08], the authors used a technique, namely *advanced reliable transmission* (see Appendix A.3), to reduce the CC of their PSMT protocol by $O(n)$. We wonder if the advance reliable transmission can be generalized in the general adversary model to further reduce the CC of our protocols to $O(h\ell\rho)$. Moreover, we conjecture that $\Omega(h\ell\rho)$ is the lower bound on CC for secure transmission of ℓ field elements. This conjecture may be proven using a similar technique as that in [SNR04, FFGS07].

Furthermore, for multicast graphs, we gave the necessary and sufficient conditions for SMT in the threshold model in such a manner that a threshold t is considered as a set in a special adversary structure $\mathcal{A}^T$. This setting is obviously general, but indirect in notation. Indeed, when considering the threshold model, the network connectivities should be more straightforward (e.g., $n > t$). Thus a more specified threshold model can be defined, with respect to which efficient protocols can then be designed. For example, if we assume that with any t nodes the adversary can control, it can completely eavesdrop on t' paths, then due to Corollary 5.4.1, there must be $n > 2t + t'$ node-disjoint paths between S and R for PSMT. Apparently, a more sophisticated definition of the threshold model is needed to simplify the conditions for APSMT.

As discussed in Section 2.2.2, an undirected multicast graph can be used to model a hypergraph in which all links are undirected, and vice versa. Similarly, a directed multicast graph, which enables one-way or mixed communication between the neighbours, can model all the multicast networks that a hypergraph can. It will be interesting to formalize this network model, and study minimal network connectivities in it, thus all kinds of secure multicast communications can be achieved.

Finally, all our protocols in this thesis use (critical) paths between S and R for message transmission. In [FY95, DWB05], the authors showed that *local interactions* between each node and its neighbours can be used for private communication against a *passive* adversary. This kind of communication does not have to respect the paths, and can be used in both point-to-point and multicast networks and in any adversary model. More importantly, it achieves a CC that is guaranteed to be polynomial in the size of the network graph. A question to be set is whether by using such local interactions, efficient RMT and SMT protocols, of which the CC is polynomial in the size of the graph, can be designed against an *active* adversary.

Bibliography

[ACdH06] S. Agarwal, R. Cramer, and R. de Hann. Asymptotically optimal two-round perfectly secure message transmission. In *Proc. CRYPTO '06*, volume 4117 of *LNCS*, pages 394–408, 2006.

[BD91] E. F. Brickell and D. M. Davenport. On the classification of ideal secret sharing schemes. *J. Cryptology*, 4(2):123–134, 1991.

[BD04] M. Burmester and Y. Desmedt. Is hierarchical public-key certification the next target for hackers? *Commun. ACM*, 47(8):68–74, 2004.

[Bei96] A. Beimel. *Secure schemes for secret sharing and key distribution.* PhD thesis, Technion, Haifa, 1996.

[BL88] J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In *Proc. CRYPTO '88*, volume 403 of *LNCS*, pages 27–35, 1988.

[Bla79] G. R. Blakley. Safeguarding cryptographic keys. In *Proc. AFIPS '79 National Computer Conference*, volume 48, pages 313–317, 1979.

[BOGW88] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computing. In *Proc. ACM STOC '88*, pages 1–10, 1988.

[BSSV97] C. Blundo, A. De Santis, R. De Simone, and U. Vaccaro. Tight bounds on the information rate of secret sharing schemes. *Des. Codes Cryptography*, 11(2):107–122, 1997.

[BW98] D. Beaver and A. Wool. Quorum-based secure multi-party computations. In *Proc. CRYPTO '98*, volume 403 of *LNCS*, pages 25–35, 1998.

[CCD88] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *Proc. ACM STOC '88*, pages 11–19, 1988.

[CDM00] R. Cramer, I. Damgård, and U. M. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *Proc. Eurocrypt '00*, volume 1807 of *LNCS*, pages 316–334, 2000.

[Csi97] L. Csirmaz. The size of a share must be large. *J. Cryptography*, 10(4):223–231, 1997. A Preliminary version published in 1995.

[CSV93] R. M. Capocelli, A. De Stantis, and U. Vaccaro. On the size of shares for secret sharing schemes. *J. Cryptology*, 6(3):157–167, 1993.

[DDFN07] I. Damgård, Y. Desmedt, M. Fitzi, and J. B. Nielsen. Secure protocols with asymmetric trust. In *Proc. Asiacrypt '07*, volume 4833 of *LNCS*, pages 357–375, 2007.

[DDWY93] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *J. ACM*, 40(1):17–47, 1993.

[Des05] Y. Desmedt. Unconditionally private and reliable communication in an untrusted network. In *Proc. IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security*, pages 38–41, 2005.

[Des06] Y. Desmedt. A high availability internetwork capable of accommondating compromised routers. *BT Technology Journal*, 24(3):1–7, 2006.

[DESN10] Y. Desmedt, S. Erotokritou, and R. Safavi-Naini. Simple and communication complexity efficient almost secure and perfect secure message transmission schemes. In *Proc. Africacrypt '10*, volume 6055 of *LNCS*, pages 166–183, 2010.

[Dol82] D. Dolev. The Byzantine generals strike again. *J. Algorithms*, 3(1):14–30, 1982.

[DW02] Y. Desmedt and Y. Wang. Perfectly secure message transmission revisited. In *Proc. Eurocrypt '02*, volume 2332 of *LNCS*, pages 502–517, 2002.

[DWB05] Y. Desmedt, Y. Wang, and M. Burmester. A complete characterization of tolerable adversary structures for secure point-to-point transmissions without feedback. In *Proc. ISAAC '05*, volume 3827 of *LNCS*, pages 277–287, 2005.

[FFGS07] M. Fitzi, M. K. Franklin, J. A. Garay, and H. V. Simhadri. Towards optimal and efficient perfectly secure message transmission. In *Proc. TCC '07*, volume 4392 of *LNCS*, pages 311–322, 2007.

[FHM99] M. Fitzi, M. Hirt, and U. Maurer. General adversaries in unconditional multi-party computation. In *Proc. Asiacrypt '99*, volume 1716 of *LNCS*, pages 232–246, 1999.

[FW98] M. K. Franklin and R. Wright. Secure communication in minimal connectivity models. In *Proc. Eurocrypt '98*, volume 1403 of *LNCS*, pages 346–360, 1998.

[FW00] M. K. Franklin and R. Wright. Secure communication in minimal connectivity models. *J. Cryptology*, 13(1):9–30, 2000.

[FY95] M. K. Franklin and M. Yung. Secure hypergraphs: Privacy from partial broadcast. In *Proc. ACM STOC '95*, pages 36–44, 1995.

[GGL98] O. Goldreich, S. Goldwasser, and N. Linial. Fault-tolerant computation in the full information model. *SIAM J. Comput*, 27(2):506–544, 1998.

[GMS74] E. Gilbert, F. MacWilliams, and N. Sloane. Codes which detect deception. *The BELL System Technical Journal*, 53(3):405–424, 1974.

[GRR98] R. Gennaro, M. O. Rabin, and T. Rabin. Simplified VSS and fact-track multiparty computations with applications to threshold cryptography. In *Proc. ACM PODC '98*, pages 101–111, 1998.

[HM00] M. Hirt and U. M. Maurer. Player simulation and general adversary structures in perfect multiparty computation. *J. Cryptology*, 13(1):31–60, 2000.

[HP06] W. C. Huffman and V. Pless. *Fundamentals of error-correcting codes.* Cambridge University Press, 2006.

[ISN87] M. Ito, A. Saito, and T. Nishizeki. Secret sharing schemes realizing general access structure. In *Proc. IEEE Globecom '87*, pages 99–102, 1987.

[KGSR02] M. Kumar, P. Goundan, K. Srinathan, and C. P. Rangan. On perfectly secure communication over arbitrary networks. In *Proc. ACM PODC '02*, pages 293–202, 2002.

[KOS+93] K. Kurosawa, K. Okada, K. Sakano, W. Ogata, and S. Tsujii. Nonperfect secret sharing schemes and matroids. In *Proc. Eurocrypt '93*, volume 765 of *LNCS*, pages 126–141, 1993.

[KS08] K. Kurosawa and K. Suzuki. Truly efficient 2-round perfectly secure message transmission scheme. In *Proc. Eurocrypt '08*, volume 4965 of *LNCS*, pages 324–340, 2008.

[KS09a] K. Kurosawa and K. Suzuki. Almost secure (1-round, *n*-channel) message transmission scheme. In *Proc. ICITS '07*, volume 4883 of *LNCS*, pages 99–112, 2009.

[KS09b] K. Kurosawa and K. Suzuki. Almost secure (1-round, *n*-channel) message transmission scheme. *IEICE Transactions*, 92-A(1):105–112, 2009.

[KS09c] K. Kurosawa and K. Suzuki. Truly efficient 2-round perfectly secure message transmission scheme. *IEEE Transaction on Information Theory*, 55(11):5223–5332, 2009.

[KW93] M. Karchmer and A. Wigderson. On span programs. In *Proc. IEEE Structure in Complexity Theory*, pages 102–111, 1993.

[Mau06] U. Maurer. Secure multi-party computation made simple. *Discrete Applied mathematics*, 154(2):370–381, 2006.

[MS78] F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes.* North-Holland Publishing Company, 1978.

[MS81] R. J. McEliece and D. V. Sarwate. On sharing secrets and Reed-Solomon codes. *Commun. ACM*, 24(9):583–584, 1981.

[NAS11] M. Nayak, S. Agrawal, and K. Srinathan. Minimal connectivity for unconditionally secure message transmission in synchronous directed networks. In *Proc. ICITS '11 (to appear)*, 2011.

[PCR09] A. Patra, A. Cloudhary, and C. P. Rangan. Brief announcement: perfectly secure message transmission in directed networks re-visited. In *Proc. ACM PODC '09*, pages 278–279, 2009.

[PCR10] A. Patra, A. Choudhary, and C. P. Rangan. On communication complexity of secure message transmission in directed networks. In *Proc. ICDCN '10*, volume 5935 of *LNCS*, pages 42–53, 2010.

[PCSR06] A. Patra, A. Choudhary, K. Srinathan, and C. P. Rangan. Constant phase bit optimal protocols for perfectly reliable and secure message transmission. In *Proc. Indocrypt '06*, volume 4329 of *LNCS*, pages 221–235, 2006.

[PSC$^+$07] A. Patra, B. Shankar, A. Choudhary, K. Srinathan, and C. P. Rangan. Perfectly secure message transmission in directed networks tolerating threshold and non threshold adversary. In *Proc. CANS '07*, volume 4856 of *LNCS*, pages 80–101, 2007.

[Rab94] T. Rabin. Robust sharing of secrets when the dealer is honest or cheating. *J. ACM*, 41(6):1089–1109, 1994.

[RBO89] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proc. ACM STOC '89*, pages 73–85, 1989.

[RS60] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *J. SIAM*, 8(2):300–304, June 1960.

[SAA96] H. Sayeed and H. Abu-Amara. Efficient perfectly secure message transmission in synchronous networks. *Inf. Comput.*, 126(1):53–61, 1996.

[SGSR08] B. Shankar, P. Gopal, K. Srinathan, and C. P. Rangan. Unconditionally reliable message transmission in directed networks. In *Proc. SODA '08*, pages 1048–1055, 2008.

[Sha79] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[Sho97] P. Shor. Polynomial time algorithms for prime factorization and discrete logarithms on a Quantum computer. *SIAM J. Comput*, 26(5):1484–1509, 1997.

[SNR04] K. Srinathan, A. Narayanan, and C. P. Rangan. Optimal perfectly secure message transmission. In *Proc. CRYPRO '04*, volume 3152 of *LNCS*, pages 545–561, 2004.

[SR06] K. Srinathan and C. P. Rangan. Possibility and complexity of probabilistic reliable communications in directed networks. In *Proc. ACM PODC '06*, pages 265–274, 2006.

[vD95] M. van Dijk. On the information rate of perfect secret sharing schemes. *Des. Codes Cryptography*, 6(2):143–169, 1995.

[WD99] Y. Wang and Y. Desmedt. Secure communication in broadcast channels: the answer to Franklin and Wright's question. In *Proc. Eurocrypt '99*, volume 1592 of *LNCS*, pages 446–458, 1999.

[WD01] Y. Wang and Y. Desmedt. Secure communication in multicast channels: the answer to Franklin and Wright's question. *J. Cryptology*, 14(2):121–135, 2001.

[WD08] Y. Wang and Y. Desmedt. Perfectly secure message transmission revisited. *IEEE Transaction on Information Theory*, 54(6):2582–2595, 2008.

[Yao79] A. C. Yao. Some complexity questions related to distributive computing. In *Proc. ACM STOC '79*, pages 209–213, 1979.

[YD09] Q. Yang and Y. Desmedt. Cryptanalysis of secure message transmission protocols with feedback. In *Proc. ICITS '09*, volume 5973 of *LNCS*, pages 159–176, 2009. Full version available in Cryptology ePrint Archive: Report 2009/632.

[YD10] Q. Yang and Y. Desmedt. General perfectly secure message transmission using linear codes. In *Proc. Asiacrypt '10*, volume 6477 of *LNCS*, pages 448–465, 2010.

[YD11] Q. Yang and Y. Desmedt. Secure communication in multicast graphs. In *Proc. Asiacrypt '11*, 2011. To appear.

Appendix A

Existing Techniques

Here we show several existing techniques in the literature. These techniques appeared in different papers that used different models and definitions. To make these results understandable, we present them using the models of Chapter 2.

A.1 Critical Identifying Algorithm by Kumar et al.

Here we present Kumar et al.'s algorithm for identifying critical paths [KGSR02]. Assume that S and R are $2\mathcal{A}$-connected in an undirected graph $G(V, E)$, the following algorithm shows how a critical set of paths P is constructed.

Critical Identifying Algorithm

```
Inputs: $G(V,E)$, $\mathcal{A} = \{A_1, \dots, A_z\}$, $S$ and $R$.

    Set $P := \emptyset$ and $i := 1$.
    While $i \leq z - 1$ do
        Set $j := i + 1$.
        While $j \leq z$ do
            If $A_i \cup A_j$ cuts all paths in $P$ then
                Select at random a path $p$ that are not cut by $A_i \cup A_j$, and
                set $P := P \cup \{p\}$.
            End if
        End while
    End while

Output: critical set $P$.                                        End.
```

Kumar et al. [KGSR02] claimed that this algorithm has the following properties:

1. The algorithm takes as in put $G(V, E)$, $\mathcal{A}$, S and R.

2. The algorithm outputs a set of paths between S and R in G, denoted by P.
3. The algorithm runs in time polynomial in $|V|$ and $|\mathcal{A}|$; i.e., $O(|V| \cdot |\mathcal{A}|^4)$.
4. The number of paths in P is polynomial in $|\mathcal{A}|$; i.e., $O(|\mathcal{A}|^2)$.[1]
5. Any solution using P is also a solution that uses all the paths between S and R.

A.2 Randomness Extractor by Srinathan et al.

In [SNR04], Srinathan et al. designed a randomness extractor to generate secret randomness's. Their design uses a Vandermonde matrix, which is not easy to present. Since Shamir's SSS essentially produces a Vandermonde matrix, in [ACdH06, KS08], the authors simplified the design of the randomness extractor by using Shamir's SSS, which uses a random polynomial. Here we show the simplified randomness extractor.

Randomness Extractor

Precondition: given w random elements $r_1, \ldots, r_w \in \mathbb{F}$, the adversary has no knowledge on ℓ of them, where $\ell < w$.

1. Using Lagrange interpolation, find a polynomial $f(x)$ of degree $deg\ f(x) = w - 1$ such that $f(i) = r_i$ for each $1 \leq i \leq w$. Note that due to the well-known unisolvence theorem of polynomial interpolation, the polynomial $f(x)$ is unique because $deg\ f(x) = w - 1$.
2. Computes $z_j = f(w + j)$ for each $1 \leq j \leq \ell$.

Postcondition: the adversary has no knowledge on $z_1, \ldots, z_\ell$. **End.**

It is easy to prove the postcondition. Indeed, because this randomness extractor essentially corresponds to a Shamir's $(w, w + \ell, 0)$-SSS [Sha79], knowing $w - \ell$ shares, any other ℓ shares remain secret and independent.

A.3 Advanced Reliable Transmission by Srinathan et al.

Consider the threshold model. In an undirected graph, S and R are connected by $n = 2t + 1$ node-disjoint paths $p_1, \ldots, p_n$. Suppose that S wants to reliably send $w = O(n)$ field elements to R via these paths. The normal solution is that S *broadcasts* the w elements via all the n paths, and R then reliably recovers these elements using majority vote. The communication complexity (CC) of this basic reliable transmission is $O(n^2\rho)$. In [SNR04], Srinathan et al. showed that the CC for reliable transmission can be reduced in some cases. In particular, they showed that if R knows the location

[1] Following our result in Observation 3.3.1 in Section 3.3, we can easily see that this claim is incomplete.

of $t' < t$ paths that are corrupted, then S can reliably send $t'+1$ elements to R with CC $O(n\rho)$. This protocol was also used in [ACdH06, KS08]. Here we show this advanced reliable transmission as follows.

Advanced Reliable Transmission Protocol for $t'+1$ elements $a_1, \dots, a_{t'+1}$

Round 1 - S to R:

1. Using Lagrange interpolation, S finds a polynomial $f(x)$ of degree $deg\ f(x) = t'$ such that $f(i) = a_i$ for each $1 \leq i \leq t'+1$.
2. For each $1 \leq i \leq n$, S sends $f(i)$ to R via path p_i.

Recovery Phase Regarding $f(x)$ as a Shamir's $(t'+1, n, 0)$-SSS, R receives n shares $f(1), \dots, f(n)$ from the n paths. R knows which t' paths are corrupted, so the corresponding t' shares are not considered. Thus the remaining shares can be seen as shared by a $(t'+1, n-t', 0)$-SSS, and $t-t'$ of them are corrupted. As shown in Section 2.7, a $(t'+1, n-t', 0)$-SSS can correct $\lfloor \frac{n-t'-t'-1}{2} \rfloor = \lfloor \frac{2t+1-2t'-1}{2} \rfloor = t - t'$ errors. Thus R can correct all the errors, and correctly recover all the shares, including the $t'+1$ elements $a_1, \dots, a_{t'+1}$, which are intended to be transmitted.
End.

Obviously, the CC of this protocol is $O(n\rho)$, which is more efficient than the ordinary broadcast of which the CC is $O(n^2\rho)$. In [SNR04, ACdH06, KS08], this technique has been used to reduce the CC of their protocols by $O(n)$.

A.4 Reliable Transmission by Desmedt et al.

In the general adversary model, RMT is possible if S and R are $2\mathcal{A}$-connected. Reliability is easy to achieve by Desmedt et al.'s protocol [DWB05], which we describe as follows.

Reliable Transmission Protocol for a single message m

Round 1 - S to R: S sends m to R via all paths in the critical set P.

Recovery Phase R finds a set $A_f \in \mathcal{A}$ such that all elements received on the paths in $P \setminus P_f$ are the same. Let m' be the element received on the paths in $P \setminus P_f$, R outputs m' as the message.

We show this protocol achieves perfect reliability. We assume that the adversary chooses a set $A_e \in \mathcal{A}$ to control. First, it is straightforward that such A_f exists, at least when $A_f = A_e$ so all the paths in $P \setminus P_f$ are uncorrupted. Next, we prove that $m' = m$. Assume that $m' \neq m$, then all the paths in $P \setminus P_f$ are corrupted. That is, $P_e \cup P_f = P$. This contradicts the $2\mathcal{A}$-connectivity.

A.5 Sub-Protocol Reconstruction by Patra et al.

Here we describe and analyse Patra et al.'s sub-protocol reconstruction scheme proposed in [PSC$^+$07]. In their work, to simplify the study of PSMT in directed graphs, the authors gave the following lemma.

Lemma A.5.1. (following [PSC$^+$07]) *Given a directed graph $G(V, E)$ where $S, R \in V$, let $\mathcal{A}$ be an adversary structure on $V \setminus \{S, R\}$ and $A_e \in \mathcal{A}$ be any set that the adversary can control.* $(0,0)$*-SMT from S to R is possible if for any subset $\mathcal{B} \subseteq \mathcal{A}$ such that $|\mathcal{B}| = 3$ and $A_e \in \mathcal{B}$,*[2] *there exists a* $(0,0)$*-SMT sub-protocol from S to R tolerating $\mathcal{B}$.*

Proof. This lemma can be proven using induction. Let $z = |\mathcal{A}|$, the induction has been shown in [PSC$^+$07].

First, if the condition is satisfied, i.e., a $(0,0)$-SMT sub-protocol exists for sub-structures of size 3, then for any sub-structure $\mathcal{B} \subseteq \mathcal{A}$ such that $|\mathcal{B}| = 4$ and $A_e \in B$, $(0,0)$-SMT is possible tolerating $\mathcal{B}$. This can be done by dividing $\mathcal{B} = \{A_1, A_2, A_3, A_4\}$ into $\mathcal{B}_1 = \{A_1, A_2, A_3\}$, $\mathcal{B}_2 = \{A_1, A_2, A_4\}$, $\mathcal{B}_3 = \{A_1, A_3, A_4\}$ and $\mathcal{B}_4 = \{A_2, A_3, A_4\}$. If the corrupted set $A_e \in \mathcal{B}$, then A_e must be in three of the divided sets $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4$. Now the message is shared by a $(2,4,0)$-SSS and each share is transmitted using a sub-protocol tolerating each $\mathcal{B}_i$ for $1 \le i \le 4$. Due to the condition, a $(0,0)$-SMT sub-protocol exists tolerating $\mathcal{B}_i$ if $A_e \in \mathcal{B}_i$, thus at least 3 shares are received 0-reliably and at most 1 share can be learned and corrupted. Since a $(2,4,0)$-SSS can correct $\lfloor \frac{4-1-1}{2} \rfloor = 1$ error (see Section 2.7), the message can be recovered with $(0,0)$-security.

Applying the above procedure again, we can find that if $|\mathcal{B}| = 5$ and $A_e \in \mathcal{B}$, then $(0,0)$-SMT is possible tolerating $\mathcal{B}$. This is because $\mathcal{B}$ can be divided into four sub-structures of size 4 such that every set in $\mathcal{B}$ is in at least three of the four divided sub-structures. Using induction, it is easy to show that when $|\mathcal{B}| = z$, which implies $\mathcal{B} = \mathcal{A}$, $(0,0)$-SMT is possible. □

Now we analyse their sub-protocol reconstruction scheme. Given an adversary structure $\mathcal{A}$ of size $z \ge 4$, $\mathcal{A}$ can be divided into four sub-structures, each of size $\lceil \frac{3z}{4} \rceil$, such that every set in $\mathcal{A}$ appears in at least three of them. Each of these subsets can be further divided in the same way, and this division continues until all the divided sub-structures are of size 3. Therefore, this scheme can otherwise be seen as a tree, in which the adversary structure $\mathcal{A}$ is the root. The nodes are the sub-structures of $\mathcal{A}$. Each parent in this tree, say of size y, has exactly four children, each of which is a sub-structure of its parent and of size $\lceil \frac{3y}{4} \rceil$. The leaves are sub-structures of size 3.

Now the sender S uses this tree. It works as that the root shares the message using a $(2,4,0)$-SSS to its children, and following this, each parent on each level shares its share

[2] In [PSC$^+$07], $\widehat{\mathcal{B}}$, which is the basis of the sub-structure $\mathcal{B}$ (see Section 1.2 for definition), is used. However this makes no difference to their proof and our analysis, so we discuss $\mathcal{B}$ instead of $\widehat{\mathcal{B}}$ here for a clearer presentation.

using a $(2, 4, 0)$-SSS to its children. At the lowest level, each leaf has a *final share*. S uses a $(0, 0)$-SMT sub-protocol (such sub-protocol exists due to Lemma A.5.1) to transmit each final share to R tolerating the corresponding leaf, which is the sub-structure of size 3. Upon the received final shares, the receiver R goes backwards in the tree. That is, R uses the above mentioned error-correcting method to reconstruct the share of each parent of the leaves, and does the same from low levels to high levels until it recovers the shared message.

Therefore, one only needs to design a $(0, 0)$-SMT sub-protocol tolerating each leaf, which is a sub-structure of size 3. The number of times that such a sub-protocol needs to be executed is the same as the number of leaves. We now calculate the number of leaves. Let μ be the height of the tree, since each parent has exactly four children, there are 4^μ leaves in this tree, and hence the $(0, 0)$-SMT sub-protocol needs to be executed 4^μ times. Next we calculate μ. From how the tree is constructed, we have the following:

$$\begin{aligned}
&\underbrace{\Big\lceil\frac{3}{4}\Big\lceil\frac{3}{4}\cdots\Big\lceil\frac{3}{4}}_{\mu} z\Big\rceil\cdots\Big\rceil\Big\rceil = 3\\
\Rightarrow&\left(\frac{3}{4}\right)^{\mu} z \le 3\\
\Rightarrow&\mu \ge \log_{\frac{4}{3}}\frac{z}{3}.
\end{aligned}$$

Thus the number of $(0, 0)$-SMT sub-protocol executions is $4^\mu \ge 4^{\log_{\frac{4}{3}}\frac{z}{3}}$, which is quasi-polynomial in z. In [YD09], Yang and Desmedt used this sub-protocol reconstruction scheme to design a $(0, 0)$-SMT sub-protocol tolerating a subset of size 3. The round complexity (RC) of their sub-protocol is 12 and the communication complexity (CC) is $O(\rho)$. Thus after running all the sub-protocols, both the RC and CC are quasi-polynomial in z.

A.6 Pseudo-basis and Pseudo-dimension by Kurosawa and Suzuki

In this section, we describe the idea of pseudo-basis and pseudo-dimension in the threshold model. This was first introduced by Kurosawa and Suzuki in [KS08]. A perfectly clear and detailed discussion of this idea has been presented in their journal paper [KS09c]. Here we briefly show this idea using our model.

Given an undirected graph with $n = 2t + 1$ node-disjoint paths $p_1, \ldots, p_n$ between a sender S and a receiver R, the messages are shared using Shamir's $(t + 1, n, 0)$-SSS. That is, given a secret s, the sender S chooses a random polynomial $f(x) = s + a_1x + \ldots + a_tx^t$, and sends $f(i)$ via path p_i for each $1 \le i \le n$. As discussed in Section 2.7, Shamir's scheme corresponds to a special case of Reed-Solomon codes. Thus the vector of shares $(f(1), \ldots f(n))$ is a codeword. That is, let C be a Reed-Solomon code, then

$(f(1), \dots f(n)) \in C$. Since at most t paths are corrupted, regarding this codeword, R receives an n-vector $\mathbf{x} = (x_1, \dots, x_n)$ with at most t errors. That is, $|\{i | x_i \neq f(i) : 1 \leq i \leq n\}| \leq t$. Let $\mathbf{c} \in C$ be the codeword sent by S, it was assumed that the errors are caused by an *error vector* $\mathbf{e} = \{e_1, \dots, e_n\}$ such that $\mathbf{x} = \mathbf{c} + \mathbf{e}$. Let E be an *error locator* such that $E = \{i | e_i \neq 0\}$. Then it is straightforward that $|E| \leq t$.

Now if w codewords $\mathbf{c}_1, \dots, \mathbf{c}_w$, regarding w messages, are sent in this way, let $\mathbf{x}_1, \dots, \mathbf{x}_w$ be the received n-vectors, then for each $1 \leq i \leq w$, we have $\mathbf{x}_i = \mathbf{c}_i + \mathbf{e}_i$, where $\mathbf{e}_i$ is an error vector. Considering a static adversary, all the error locators $E_1, \dots, E_w$ together indicate at most t error locations. That is, the dimension of the vector space spanned by $\mathbf{e}_1, \dots, \mathbf{e}_w$ is at most t.

In [KS09c, Lemma 1], the authors showed the following fact: given $1 \leq i \leq w$ and $1 \leq g_1 < \dots < g_b \leq w$, if there exist $a_1, \dots, a_b \in \mathbb{F}$ such that

$$\mathbf{x}_i + a_1\mathbf{x}_{g_1} + \dots + a_b\mathbf{x}_{g_b} \in C,$$

then

$$\mathbf{e}_i + a_1\mathbf{e}_{g_1} + \dots + a_b\mathbf{e}_{g_b} = 0.$$

Here if $\mathbf{x}_i + a_1\mathbf{x}_{g_1} + \dots + a_b\mathbf{x}_{g_b} \in C$, then $\mathbf{x}_i$ is said to be *linearly pseudoexpressed* by $\{\mathbf{x}_{g_1}, \dots, \mathbf{x}_{g_b}\}$. In [KS08, KS09c, Fig. 2], an efficient way to check if a vector is linearly pseudoexpressed by a set of vectors has been given.

Now the *pseudo-basis* $B = \{x_{g_1}, \dots, x_{g_b}\}$ is defined such that every n-vector $\mathbf{x}_i$ $(1 \leq i \leq w)$ is linearly pseudoexpressed by B. From the above fact, we have that every error vector $\mathbf{e}_i$ is in the linear span of $\mathbf{e}_{g_1}, \dots, \mathbf{e}_{g_b}$. Thus it is clear that the corresponding error locators $E_{g_1}, \dots, E_{g_b}$ indicate all the error locations during the transmission. As discussed earlier, the dimension of the vector space spanned by $\mathbf{e}_1, \dots, \mathbf{e}_w$ is at most t. That is, at most t error vectors are needed to span this vector space. Thus $|\{\mathbf{e}_{g_1}, \dots, \mathbf{e}_{g_b}\}| \leq t$, and hence we have the *pseudo-basis* $|B| \leq t$.

During the transmission, if the receiver R constructs a pseudo-basis from the received n-vectors and reliably sends it back to S, then S is able to find all the error locations for R. With all the error locations known, R can finally recover all the w messages with the correct elements.