Secure Schemes for Secret Sharing and Key Distribution

Amos Beimel

Secure Schemes for Secret Sharing and Key Distribution

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Science

Amos Beimel

Submitted to the Senate of the Technion – Israel Institute of Technology
Sivan, 5756 Haifa June 1996

The work described herein was supervised by prof. Benny Chor under the auspices of the computer science committee.

I always enjoy thanking people that helped me:

Warm thanks to Benny Chor for his devoted supervision, and for everything I learned from him. Benny taught me to be sceptic about proofs, and suspect that they might contain errors and gaps. Above all, I learned from him that research not only involves proving theorems, but trying to understand why things are true. I enjoyed being his student.

To Eyal Kushilevitz for our joint work, from which I gained a lot. Thanks for the fact that I could always consult with him, to bother him, and go drink coffee with him.

To all my coauthors who cooperated with me during this research: Benny Chor in [10, 9] which appears in Chapter 6 and Chapter 7, Anna Gál and Mike Paterson in [11] which appears in Chapter 5, and Mike Burmester, Yvo Desmedt, and Eyal Kushilevitz in [7] which appears in Chapter 8. I enjoyed working with them all.

To all the colleagues who I consulted during this research: Carlo Blundo, Shimon Even, Oded Goldreich, Amir Herzberg, Hugo Krawczyk, Noam Nisan, Erez Petrank, Avi Wigderson and Moti Yung.

To all my friends, lecturers and secretaries in the computer science department at the Technion for the lovely ten years I spent here.

Finally, to my parents Zelda and Chaim and my family for all their help and support.

The generous financial help of the Technion is gratefully acknowledged.

Contents

Contents - (Continuation)

List of Figures

Abstract

In recent years the security of operations taking place over a computer network become very important. It is necessary to protect such actions against "bad" users who may try to misuse the system (e.g. steal credit card numbers, execute actions without authorization, read personal mail, or impersonate other users). Many cryptographic protocols and schemes were designed to solve problems of this type. This thesis deals with two fundamental cryptographic tools that are useful in such contexts: generalized secret sharing schemes, and key distribution schemes.

Both secret sharing schemes and key distribution schemes are used in multi-party systems. Secret sharing schemes enable some predetermined sets of parties to reconstruct a given secret. These schemes make it possible to store secret information in a network, such that only "good" (for example, large enough) subsets can reconstruct the information. Furthermore, by using these schemes we can enable only "good" subsets to perform actions in a system (e.g. sign a check). The secret sharing schemes that have received most attention are threshold secret sharing schemes. These are schemes in which the sets which can reconstruct the secret are those with of cardinality greater than a certain threshold. In this work we consider these schemes, as well as generalized secret sharing schemes. In the generalized schemes the subsets capable of reconstructing the secret constitute an arbitrary monotone collection. The second type of schemes we consider are *key distribution schemes* which enable every subset of parties to generate a secret key (different subsets have different keys). These keys can be used, for example, in private key cryptosystems or for authentication.

In this thesis we consider (bad) parties which have unlimited power, i.e. we consider the information theoretic setting (in contrast to the setting in which parties are limited to probabilistic polynomial time computations). For the two types of schemes we assume that there exists an off-line dealer which distributes private pieces of information to the parties when the system is initialized. Since the security of the schemes is based on the secrecy of these pieces of information, the pieces should be kept on a secure (and expensive) storage medium. Therefore, minimizing the amount of information distributed to the parties is significant for practical usages. Most of our research is concerned with the (space) efficiency of the schemes, which is measured by the size of the of pieces (i.e. the length of their binary representation). Many of the existing schemes require long pieces, and their size is exponential in the number of parties in the system (e.g. the secret sharing schemes

of [16, 105]). Our main goal in the thesis is to understand when these schemes are inherently exponential, and when they can be made more efficient.
In this work we consider four topics:

Communication in Key Distribution Schemes: We study the relationships between communication and space efficiency of key distribution schemes. We prove that communication does not help in *unrestricted schemes*. On the other hand, we show that for restricted schemes, which are secure only when used by a limited number of conferences, communication can substantially improve the space efficiency. Furthermore, we prove lower bounds on the space efficiency of restricted schemes.

Linear Secret Sharing Schemes and Monotone Span Programs: In the best secret sharing schemes known to date for most access structures, the size of the pieces is exponential in the number of parties in the system. No matching lower bounds are known. Therefore, we restricted our attention to the class of linear secret sharing schemes. This class of secret sharing schemes contains essentially all known secret sharing schemes. There is a close relation between linear secret sharing schemes and a linear algebraic model of computation called *span programs*. The existence of an efficient linear schemes for an access structure is equivalent to the existence of a small monotone span programs for the characteristic function of the access structure. In this work we prove $\Omega(n^{2.5})$ lower bound on the size of pieces in linear secret sharing schemes (and monotone span programs) for an explicit access structure.

Secret Sharing with Public Reconstruction: All known constructions of information theoretic t-out-of-n secret sharing schemes require *secure, private* communication channels among the parties for the reconstruction of the secret. We investigate the cost of performing the reconstruction over *public* communication channels. A naive implementation of this task distributes $O(n)$ one time pads to each party. This results in pieces whose size is $O(n)$ times the secret size. We present several implementations of such schemes that are substantially more efficient: (1) A scheme enabling multiple reconstructions of the secret by different subsets of parties, with factor $O(n/t)$ increase in the pieces' size. (2) A one-time scheme, enabling a single reconstruction of the secret, with $O(\log(n/t))$ increase in the pieces' size. We prove that the first implementation is optimal (up to constant factors) by showing a tight $\Omega(n/t)$ lower bound for the increase in the pieces' size.

Computing Functions of a Shared Secret: We introduce and study threshold (t-out-of-n) secret sharing schemes with respect to a *family of functions* $\mathcal{F}$. Such schemes allow any set of at least t parties to reconstruct privately the value $f(s)$ of a (previously distributed) secret s (for any $f \in \mathcal{F}$). Smaller sets of players "know nothing" about the secret. The goal is to make the pieces as short as possible.

Notation

P_i	–	The i-th party in the system.
n	–	The number of parties in the system.
R	–	The domain of random strings.
r	–	A random string.
$\mathcal{K}$	–	A field.
$\mathrm{GF}(q)$	–	The finite field with q elements.
G	–	A reconstructing ("Good") set.
B	–	A "Bad" (curious) set.
VIEW_B	–	The knowledge that a coalition B has.
$\mathrm{I}(X,Y)$	–	The mutual information between the random variables X and Y.
$\mathrm{H}(X)$	–	The entropy of a random variable X.
$\mathrm{span}\,(V)$	–	The linear space spanned by the vectors in V.
$\mathrm{rank}(V)$	–	The rank (dimension) of the linear space spanned by the vectors in V.
$\circ$	–	Concatenation of strings.
s	–	A secret in secret sharing schemes.
Π	–	A secret sharing scheme.
$\Pi_i(s,r)$	–	The piece (share) of P_i generated with secret s and random string r in the secret sharing scheme Π.
$\mathcal{A}$	–	An access structure.
S	–	The domain of secrets in secret sharing schemes.
t	–	The size of a reconstructing set in threshold secret sharing schemes.
ℓ	–	The length of the binary string that represents the secret.
$\mathcal{U}$	–	A key distribution scheme.
$\mathcal{U}_i(r)$	–	The piece of P_i generated from r in the key distribution scheme $\mathcal{U}$.
K	–	The domain of keys in key distribution schemes.
k	–	A key in key distribution schemes.
b	–	Upper bound on the number of "bad" parties in key distribution schemes.
C_G	–	The communication exchanged when the set G reconstructed the key.
g	–	The size of a reconstructing set in key distribution schemes.

$\mathcal{F}$	–	Family of functions.
$\mathcal{LIN}_\ell$	–	The family of linear functions on GF(2^ℓ).
$e_i(x)$	–	The function that returns the i-th bit of x.
$\mathcal{BIT}_\ell$	–	The family of bit functions $\{e_1, \ldots, e_\ell\}$.
$\mathcal{ALL}_\ell$	–	The family of all functions with domain $\{0,1\}^\ell$.
XOR	–	The 2-out-of-2 secret sharing scheme $\mathrm{XOR}(s,r) = \langle r, s+r \rangle$.
$\vec{1}$	–	A vector in which every coordinate equals 1.
$\mathrm{SP}_{\mathcal{K}}(f)$	–	The size of the smallest span program over the field $\mathcal{K}$ computing f.
$\mathrm{mSP}_{\mathcal{K}}(f)$	–	The size of the smallest monotone span program over the field $\mathcal{K}$ computing f.
$\mathcal{M}_f$	–	The family of all of the minterms of f.
T_H	–	The core of a minterm H (in Definition 5.5).
$\mathcal{H}$	–	A critical family.

Chapter 1

Introduction

In recent years the security of operations taking place over a computer network become very important. It is necessary to protect such actions against "bad" users who may try to misuse the system (e.g. steal credit card numbers, execute actions without authorization, read personal mail, or impersonate other users). Many cryptographic protocols and schemes were designed to solve problems of this type. This thesis deals with two fundamental cryptographic tools that are useful in such context: generalized secret sharing schemes, and key distribution schemes.

Both secret sharing schemes and key distribution schemes are used in a multi-party system. Secret sharing schemes enable only some predetermined sets of parties to reconstruct a given secret. These schemes make it possible to store secret information in a network, such that only "good" (for example, large enough) subsets can reconstruct the information. Furthermore, by using these schemes we can enable only "good" subsets to perform actions in a system (e.g. sign a check). The secret sharing schemes that have received most attention in the literature are threshold secret sharing schemes. These are schemes in which the sets which can reconstruct the secret are all the sets of cardinality greater than a certain threshold. In this work we consider these schemes, as well as generalized secret sharing schemes. In the generalized schemes the subsets capable of reconstructing the secret constitute an arbitrary monotone collection[1]. The second type of schemes we consider are *key distribution schemes* which enable every subset of parties to generate a secret key (different subsets have different keys). These keys can be used, for example, in private key cryptosystems or for authentication.

In this thesis we consider (bad) parties which have unlimited power, i.e. we consider the information theoretic setting (in contrast to the setting in which parties are limited to probabilistic polynomial time computations). For the two types of schemes we assume that there exists an off-line dealer which distributes private pieces of information to the parties when the system is initialized. Since the security of the schemes is based on the secrecy of

[1] A collection of sets is monotone if for every sets A and B, such that A is in the collection and $A \subseteq B$, then B is also in the collection.

these pieces of information, the pieces should be kept on a secure (and expensive) storage medium. Therefore, minimizing the amount of information distributed to the parties is significant for practical usages. Most of our research is concerned with the (space) efficiency of the schemes, which is measured by the size of the of pieces (i.e. the length of their binary representation). Many of the existing schemes require long pieces, and their size is exponential in the number of parties in the system (e.g. the secret sharing schemes of [16, 105]). Our main goal in the thesis is to understand when these schemes are inherently exponential, and when they can be made more efficient. In addition, we study whether communication and interaction reduce the space requirements. Furthermore, we investigate how the communication model (secure private channels vs. public channels) effects the space requirements. While examining this questions, it is important to distinguish between one-time schemes in which the action takes place only one time (e.g. one set will reconstruct the secret), and unrestricted schemes in which the number of times an action can take place is not restricted (e.g. many sets will reconstruct the secret). For example, we show that the space requirements of a scheme in which only one set will generate a key is substantially smaller than the space requirements of a scheme in which many sets will generate keys. In the remaining of this introduction we give informal definitions of key distribution schemes and of secret sharing schemes, and survey the results of this work.

1.1 Key Distribution Schemes

In various multi-party systems, the need for generating a secret key, common to a subset of the parties, occasionally arises. Such key can be used, for example, in establishing a secure private key cryptosystem among the members of the subset. A *non-communicating key distribution scheme* for conferences of size g which is secure against any disjoint coalition of b parties is a scheme where an off-line dealer initially distributes n pieces of information, one per party. Each party receives his piece discretely. After this initial distribution takes place, the parties can reconstruct keys in a secure fashion. That is:

1. Every conference (set) G of g parties has a common key which is called the key of the conference G. Every member of G can reconstruct the key from his piece (and the conference identity) without any communication with other parties or with the dealer. That is, the party applies some function to its piece and evaluates the key.

2. Every "bad" coalition B of at most b parties does not gain any information on the key of any disjoint conference G. That is, the pieces of B do not expose any additional knowledge on the key of G.

It might help the reader to remember that the size of conferences is denoted by g which stands for "$g$ood" parties, and the size of coalitions is denoted by b which stands for "$b$ad" parties. In these schemes, the size of the pieces is a function of the size of the keys, the

number of parties n, the size of conferences g, and the size of coalitions b. Formal definition of key distribution schemes is given in Chapter 3.

The trivial key distribution scheme chooses an independent random key for every conference G, and gives party P_i all the keys of the conferences that contain him. This scheme is secure against coalitions of size $n-g$. However, the scheme is not efficient (the size of every piece in this scheme is $\binom{n-1}{g-1}$ times size of the keys), and for smaller values of b there are more efficient schemes. Blom [24] was the first to consider information theoretic key distribution schemes. He presented an efficient non-communicating scheme, based on MDS codes, for conferences of size 2 and coalitions of size b. The size of the pieces in Blom's scheme is $b+1$ times the size of the keys (compared with $n-1$ times the size of the keys in the trivial scheme for $g=2$). Blundo et. al. [29] present key distribution schemes for larger conferences, based on symmetric multinomials. Their multinomials have g variables and degree of at most b in each variable. The size of the pieces in their scheme is $\binom{g+b-1}{b}$ times the size of the keys. For large values of g and b, this expression is quite large (e.g. for $g=b=n/2$ the size of pieces is $2^{n-o(n)}$ times the size of the keys). However, using entropy arguments, Blundo et. al. [29] prove a tight lower bound on the size of the pieces. Therefore, their scheme is space-optimal. In this work, we apply direct combinatorial arguments (no entropy) to prove the same lower bound. Our proof has two advantages. First, in our opinion, it is more intuitive and less technical. Second, it actually applies to a weaker notion of security, thereby providing a stronger result. This stronger result is used in proving our lower bound on communicating schemes, which is described in the next paragraph.

The large lower bound (for big conferences and coalitions) raises the question whether communication among the reconstructing parties could be of help in reducing the size of pieces. Just like the non-communicating schemes, we first require that even if an unrestricted number of conferences communicate in order to generate keys, these keys remain secure with respect to disjoint coalitions of size b. Since no secure channels among parties can be assumed, communication takes place via public channels. One problem which arises is that the communication of one conference could leak information on the keys of *other* conferences. Therefore, we require that even if a "bad" coalition overheard the communication of all the conferences, the coalition does not gain information on any key of a disjoint conferences. We prove that, regrettably, such unrestricted communicating schemes require pieces as large as the pieces in non-communicating schemes.

This negative result motivates the introduction of *restricted* communicating schemes. These schemes can be used only for a restricted number of conferences, whose identity is not known beforehand. We construct an efficient one-time secure scheme, where the size of the pieces is $O(b/g)$ times the size of the keys. This is a substantial improvement over the one-time communicating scheme of [29], where the size of the pieces is $g+b-1$ times the size of the keys. Following [29], our schemes are *non-interactive*; each party sends messages which depend only on his piece of information and not on messages received from other parties. We prove a nearly tight lower bound on the size of the pieces in every one time key

distribution scheme; the size of each piece is at least $b/(g-1)$ times the size of keys. Using τ copies of the one-time scheme, we construct a scheme which is secure for τ conferences. The size of pieces in this scheme is $O(\frac{\tau(b-1)}{g})$ times the size of the keys. We show that the domain of pieces of every party in a communicating key distribution scheme, which is secure for τ conferences, is at least $\max\left\{\tau, \tau^{(1-1/g)}(b-1)/g\right\}$ times the size of the keys. Hence, the dependence on τ of the domain of pieces in our scheme cannot be completely avoided, and for $\tau \geq \binom{g+b-1}{b}$ every τ-restricted scheme cannot be more efficient then the non-communicating (unrestricted) scheme.

1.2 Secret Sharing Schemes

In a generalized secret sharing scheme a dealer has a secret taken from some domain, which he wants to share among a collection $\mathcal{A}$ of subsets of the n parties. The collection $\mathcal{A}$ is called the access structure. The dealer discretely distributes private pieces of information (also called shares) to the parties, such that any subset of parties in $\mathcal{A}$ can reconstruct the secret from its pieces, while any subset not in $\mathcal{A}$ cannot reveal any partial information about the secret in the information theoretic sense (formal definition of secret sharing schemes appears in Chapter 3). A secret sharing scheme can only exist for monotone access structures. If a subset B can reconstruct the secret (i.e. $B \in \mathcal{A}$), then every superset of B can also reconstruct the secret. If the subsets that can reconstruct the secret are all the sets whose cardinality is at least a certain threshold t, then the scheme is called t-out-of-n threshold secret sharing scheme.

One of the most important issues when designing secret sharing schemes is the size of the pieces. Even with the best known schemes (e.g. [16, 105]), most general access structures require pieces of size exponential in the number of parties even if the domain of the secret is binary (the length is at least $2^{0.5n}$, where n is the number of participants). Therefore, the parties will not have enough memory to store their pieces even in fairly small networks (leaving aside the question of secure storage). The question if there exist more efficient schemes, or if there exists an access structure that does not have (space) efficient schemes remains open. We conjecture that the later is true:

Conjecture 1.1: *There exists an $\epsilon > 0$ such that for every positive integer n there is an access structure with n parties, for which every secret sharing scheme distributes pieces of length exponential in the number of parties n, that is $2^{\epsilon n}$.*

Proving (or disproving) this conjecture is one of the most important open questions concerning secret sharing.

The best lower bound that is known to date is due to Csirmaz [44, 43]. His proof gives, for every n, an explicit access structure with n parties for which the sum of the sizes of the pieces in every secret-sharing scheme is $\Omega(n^2/\log n)$ times the size of the secrets (for every finite set of possible secrets).

Proving Conjecture 1.1 was one of the main goals of this research. We did not succeed in this task. Therefore, we limited our attention to the class of linear secret sharing schemes, which contains essentially all known secret sharing schemes. We proved lower bounds on the size of pieces in linear secret sharing schemes. We discuss these schemes in Section 1.2.1. In Section 1.2.2 we discuss secret sharing schemes in which reconstruction takes place on public channels. In Section 1.2.3 we discuss secret sharing schemes in which not only the secret but functions of the secret can be reconstructed without revealing any other information on the secret.

1.2.1 Linear Secret Sharing Schemes and Monotone Span Programs

In most known secret sharing schemes every set in the access structure reconstructs the secret using a linear function of its pieces. That is, every party gets a few elements from some finite field as its piece, and every set in the access structure reconstructs the secret using a linear combination of the elements held by its parties. We call such schemes linear. For example, the following schemes are linear: Blakley's threshold scheme [22], Shamir's threshold scheme [101], Kothari's linear threshold schemes [72], Ito, Saito and Nishizeki's scheme based on decomposition of the access structure to threshold access structures [62], Benaloh and Leichter's schemes based on monotone formulas [16], Simmons, Jackson and Martin's schemes based on geometric configurations [103, 104, 105], Brickell and Davenport's schemes based on matroid representation [34, 35] (see also Karchmer and Wigderson [68] and Bertilsson and Ingemarsson [19]), and the decomposition technique schemes proposed in [27, 30, 36, 107, 108]. The survey of Stinson [106] contains a description of most of these scheme. Jackson and Martin [64] proved that linear schemes are equivalent to the geometric schemes of [103, 104, 105]. Counting arguments imply that for most access structures the size of the pieces in every linear scheme is at least $2^{n/2}$. No such bound is known for an explicit access structure.

There is a close relation between linear secret sharing schemes and a linear algebraic model of computation called *span programs* which was introduced by Karchmer and Wigderson [68]. The existence of an efficient linear schemes for a specific access structure is equivalent to the existence of a small monotone span programs for the characteristic function of the access structure. A span program for a Boolean function is presented as a matrix over some field. Every row of the matrix is labeled by a literal (a variable or a negated variable). Given an assignment $\delta \in \{0,1\}^n$ we consider the rows of the matrix whose labels are consistent with the assignment (that is, either rows labeled by some x_i such that $\delta_i = 1$ or rows labeled by some $\bar{x}_i$ such that $\delta_i = 0$). The span program accepts the assignment if and only if the all-ones row is a linear combination of these rows. The size of a span program is the number of rows in its matrix. (Definitions are given in Chapter 4). The class of functions with polynomial size span programs is equivalent to the class of functions with polynomial

size counting branching programs [37, 68]. Span program size is a lower bound on the size of symmetric branching programs [68].[2] Lower bounds for span programs also imply lower bounds for formula size.

Monotone span programs have only positive literals (non-negated variables) as labels of the rows. They compute only monotone functions, even though the computation uses non-monotone linear algebraic operations. Karchmer and Wigderson [68], following [35], proved that if there is a monotone span program for some function then there exists a linear secret sharing scheme for the corresponding access structure in which the sum of the sizes of the pieces of all the parties is the number of rows in the span program, i.e. the size of the span program. Therefore, every lower bound on the total size of pieces in a linear secret-sharing scheme is also a lower bound on the size of monotone span programs for the same function. On the other hand, lower bounds for monotone span programs imply the same lower bounds for linear secret-sharing schemes [6, 8, 52]. We will prove these connections in Chapter 4.

It is known that every function with a polynomial size span program is in NC (this follows from [18, 37, 68, 83]). The monotone analogue of this statement does not hold: Babai et. al. [4] exhibit a function that is computable by monotone span programs whose size is linear but requires super-polynomial size monotone circuits. On the other hand, the reduction in [68] from symmetric branching programs to span programs preserves monotonicity, and thus lower bounds for monotone span programs imply lower bounds for monotone symmetric branching programs and for monotone formula size. We note that if P $\not\subseteq$ non-uniform-NC then there are functions with small monotone circuit complexity that cannot be computed by polynomial size span programs. Consider the following language MVAL $= \{(C, w) : C$ is a monotone Boolean circuit that accepts $w\}$. If MVAL has a polynomial size span program then it has a polynomial size NC circuit. Since MVAL is P-complete, this would imply that P $\subseteq$ non-uniform-NC. On the other hand, with the proper representation MVAL has a small (but not shallow) monotone circuit.

The $\Omega(n^2/\log n)$ lower bound for monotone span program size, implied by the lower bound for any secret sharing scheme of [44], is the strongest previously known lower bound for an explicit function on n variables. The method presented in [44] cannot give lower bounds larger than $\Omega(n^2)$. In Chapter 5 we present a new technique for proving lower bounds for monotone span programs. Using this technique, we present an $\Omega(n^{2.5})$ lower bound for an explicit function on n variables. We obtain this bound for the Boolean function that is 1 if and only if an input graph contains a 6-clique. We present several other applications of our technique to explicit functions. A recent result [4] demonstrates that our technique can yield super-polynomial lower bounds of $n^{\Omega(\log n)}$ for monotone span programs by considering a problem in extremal set theory. It remains open whether our method could yield strictly exponential lower bounds.

[2] The model of symmetric branching programs is essentially the same as that of (undirected) contact schemes (for definitions, see Example 4.5 and [68]).

1.2.2 Secret Sharing Schemes with Public Reconstruction

When reconstructing a secret, an authorized subset of the parties collects their pieces, and uses them to reconstruct the secret. It is required that after a reconstruction, only the parties which participated in the reconstruction will know the secret. All known schemes that guarantee information theoretic secrecy require the use of secure, private communication channels between the parties that participate in the reconstruction. The question we raise is whether reconstruction can be done without assuming that the channels are secure, while maintaining the security of the schemes. We consider the scenario in which the "bad" parties can overhear any communication, so from their point of view the channels are public. On the other hand, "good" parties hear only messages sent to them. (In particular, from the point of view of the "good guys", the channels do not carry any of the potential advantages of a broadcast channel.) For simplicity, we will consider only threshold (t-out-of-n) secret sharing schemes in this introduction.

The simplest way to implement such public reconstruction securely is to hand to each party upon system initialization, in addition to his original piece, $2(n-1)$ one time pads. These pads are used in order to simulate a private channel on a public one. In the private channel scenario, reconstruction is typically done by exchanging pieces among parties. To enable such exchange with every other participant, each party will need two pads per participant: one for receiving a piece, and one for sending the piece. Thus the simple implementation results in an $O(n)$ multiplicative factor increase in the size of each piece.

We design substantially more efficient schemes of three types. The first type is *unrestricted schemes*. In these schemes, any number of authorized sets (each containing at least t parties) may reconstruct the secret, after communicating on the public channel. Any disjoint coalition of at most $t-1$ parties does not gain any partial information on the secret, given the coalition's pieces and the communication of the sets that reconstructed the secret. We describe unrestricted schemes in which the size of the pieces is $O(n/t)$ times the size of the original secret. We complement this result by proving a nearly tight n/t lower bound on the increase in the pieces' size for any unrestricted scheme. Our construction (upper bound) has the property that in order to participate in more than one reconstruction, every party that has already reconstructed the secret must store it. This is problematic in applications where an adversary might break into the computer of the secret holder. (One of the advantages of traditional secret sharing is that breaking into the computer of a "piece holder" does not compromise the secret.) The unrestricted non-interactive schemes of Section 7.4 solve this problem, but the piece size there is n times the secret size.

The second type of schemes we consider here are *one time schemes*, in which only a single authorized set (containing at least t parties) will reconstruct the secret securely. It is not known during system initialization which set will reconstruct the secret, and the dealer has to accommodate any possible set. For example, these schemes can be used to enable one time activities like the firing of a ballistic missile or the opening of a sealed safe. We describe one-time schemes in which the size of the pieces in $O(\log(n/t))$ times the original

secret size. Next, we consider one time schemes where one authorized set of size *exactly* t will reconstruct the secret. Additional parties in supersets with more than t parties jointly have enough information to reconstruct the secret. However, they cannot reconstruct the secret over the public channel, because communicating it from members of the authorized set is not possible in a secure way. This means that the authorized sets that can securely reconstruct the secret do *not* necessarily form a *monotone* access structure. We design such schemes with just $O(1)$ multiplicative increase in the piece size (for any threshold t).

In light of our results, one may wonder if the initial distribution of pieces can also be done over public channels. By the properties of "regular" schemes, each participant requires a piece whose conditional mutual information with the secret (given $t-1$ pieces) is at least the entropy of the secrets [69]. It is not possible to start with pieces of smaller conditional entropy and increase it by communicating over public channels, even if interaction is allowed [77, 2]. Thus in our model, it is necessary to have secure initial distribution of pieces from the dealer to the participants. However, from practical point of view the distribution stage is an off-line process which is typically done upon system initialization (unlike the reconstruction stage). Thus, assuming private initial distribution is reasonable.

1.2.3 Computing Functions of a Shared Secret

Suppose that we are interested in sharing a secret file among n parties in a way that will later allow any t of the parties to test whether a particular string (not known in the sharing stage) appears in this file. This test should be done without revealing the content of the whole file, or giving any other information about the file. This problem is an extension of the traditional problem of threshold (t-out-of-n) secret-sharing (over secure private channels). In this work we introduce a more general definition of t-out-of-n secret-sharing schemes with respect to a *family of functions* $\mathcal{F}$. These schemes, in addition to traditional requirements of secret sharing schemes, allow authorized sets of parties to reconstruct some information about the secret without revealing the secret itself. More precisely, sets of size at least t can evaluate $f(s)$ for any function $f \in \mathcal{F}$ in a way that after the evaluation of $f(s)$ any coalition of size less than t gets no information about the secret s which is not implied by $f(s)$. In other words, the parties in the coalition might know the value $f(s)$, but they know nothing more than that, even though they have heard all the communication during the reconstruction of $f(s)$.

Clearly, if we consider a family $\mathcal{F}$ that includes only the identity function $f(s) = s$, then we get the traditional notion of secret-sharing schemes. Simultaneous sharing of many secrets is also a special case of our setting: let $s_1, \ldots, s_\ell$ be secrets we want to share *simultaneously*. Construct the secret $s \triangleq s_1 \circ s_2 \circ \ldots \circ s_\ell$, and the functions which can be reconstructed are the functions $f_i(s) \triangleq s_i$. The question of sharing many secrets *simultaneously* was considered (with some differences in the definitions) by several researchers [26, 31, 59, 65, 66, 69, 79].

Threshold cryptography [46, 48, 49] is also a special case of secret sharing with respect

to a family of functions.[3] A typical scenario of threshold cryptography is the following: We want to enable every t parties to sign a document such that any coalition of less than t parties cannot sign any other document (even if the coalition knows signatures of some documents). To achieve this goal the key is shared such that every t parties can generate a signature from their pieces without revealing any information on the key except the signature. Specificly, assume we have a signature function $\mathrm{SIGN} : M \times K \to O$ where M is the domain of messages, K the domain of keys, and O the domain of signatures. For every $m \in M$ define $f_m : K \to O$ as $f_m(k) = \mathrm{SIGN}(m, k)$. The previous scenario is simply sharing the key with respect to the family $\{f_m : m \in M\}$. This shows that secret sharing with respect to a family of functions is a natural primitive.

Obviously, a possible solution to the problem of sharing a secret with respect to a family $\mathcal{F}$ is by sharing *separately* each of the values $f(s)$ (for any $f \in \mathcal{F}$) using known threshold schemes. While this solution is valid, it is very inefficient, in particular when $|\mathcal{F}|$ is large. Therefore, an important goal is to realize such schemes while using "small" pieces. For example, to share a single bit the average (over the parties) length of pieces is at least $\log(n - t + 2)$ (for $2 \leq t \leq n - 1$) [71] and $\log n$ bits are sufficient [101] (where n is the number of parties in the system). Therefore, the obvious solution for sharing ℓ bits simultaneously will require $\ell \log n$ bits. By [69], it can be shown that pieces of at least ℓ bits are necessary. We show that ℓ-bit pieces are also sufficient if *interactive* reconstruction on private channels is allowed, and $O(\ell)$-bit pieces are sufficient if non-interactive reconstruction (on public channels) is used[4].

We present an interactive scheme in which ℓ-bit secrets are distributed using ℓ-bit pieces; this scheme allows the reconstruction of the exclusive-or of every subset of the bits of the secret (and not only the bits themselves). We use this scheme to construct schemes for other families of functions. The length of the pieces in these schemes can be much longer than the length of the secret (this is the case for the strings and signatures examples). An interesting family of functions that we shall consider is the family of *all* functions of the secret. For this family, we construct a scheme in which the length of the pieces is $\approx 2^\ell \log n$ (where ℓ is the length of the secret and 2^ℓ is the length of the description of a function which is the input for the reconstruction). That is, the length of the pieces is linear in the description of a function in the family which is the input for the reconstruction stage. In this scheme the reconstruction requires no interaction and can be held on public channels. If we allow interaction on private channels during the reconstruction, then the length of the pieces can be reduced to 2^ℓ. We do not know whether for the family of all functions there is a scheme in which the length of the piece (which is the amount of space required by the parties) is polynomial in the length of the secret.

While considering the question of computing functions of a secret, we deal mainly with

[3]The functions considered in [49, 46] are however very limited and the scenario in [46] is restricted to computational security.

[4]In fact, both results require ℓ to be "sufficiently large": $\ell \geq \log n$ in the interactive case and $\ell \geq \log n \log\log n$ in the public channels model.

two models: the *private channels model* in which the reconstruction might require a few rounds of communication; and the *public channels model* in which the reconstruction is non-interactive. On one hand, in the private channel model, a coalition that does not intersect the reconstructing set will know nothing on $f(s)$. On the other hand, the public channels model does not require secure private channels and synchronization; hence, the reconstruction is more efficient. Unlike Section 1.2.2, while reconstructing a function over the public channels, every party in the system might learn the value of the function (but no additional information).

Interaction seems to be useful in the reconstruction stage. It enables us to reduce the length of the pieces by a factor of $\log n$. We also demonstrate this fact by studying *ideal* threshold schemes; these are schemes in which the size of the pieces equals the size of the secrets[5] (see, e.g., [8, 34, 35, 63, 100]). We deal with the characterization of the families of functions $\mathcal{F}$ that can be evaluated in an ideal threshold scheme. For the interactive private channel model, we prove that these functions are "essentially" only the *linear* functions. For the public channels model, we prove that $\mathcal{F}$ cannot even contain any Boolean function (for every family that contains the identity function).

Relation to private computation. In our schemes we require that authorized sets of parties can reconstruct a function of the secret without leaking any other information about the secret. This resembles the requirement of (n, k)-private protocols [13, 39] in which the set of all n parties can evaluate a function of their inputs in a way that no set of less than k parties will gain any additional information. Indeed, in some of our schemes a set B of size t uses a (t, t)-private protocol. However, it is not necessary to use private protocols in the reconstruction, since the parties are allowed to leak information about their inputs (the pieces) as long as they do not leak additional information about the secret. Moreover, using private protocols does not solve the problem of efficiently sharing a secret with respect to all functions, since not all functions can be computed (t, t) privately [13, 40]. Furthermore, the parties cannot use the $(t, t/2)$-private protocols of [13] or [39] since this means that coalitions of size greater than $t/2$ (but still smaller than t) gain information. However, if we only require that coalitions of at most $t/2$ parties should not gain any partial information on the secret then we can share the secret with any traditional scheme and reconstruct any function of the secret using the private protocols of [13] or [39] (since we want to evaluate a function of the secret which is a function of the pieces).

1.3 Organization

The remainder of this thesis is organized as follows. In Chapter 2 we discuss previous works on key distribution schemes and secret sharing schemes. In Chapter 3 we present the exact definitions of the models and the schemes we consider. In Chapter 4 we define linear secret sharing schemes and monotone span programs and we discuss the equivalence between

[5] By [69], the size of each piece is at least the size of the secret.

these two models. In Chapter 5 we present lower bounds for monotone span programs. In Chapter 6 we discuss the roll of communication in key distribution schemes. In Chapter 7 we deal with secret sharing with public reconstruction. In Chapter 8 we deal with sharing a family of function of a secret. Finally, in Chapter 9 we conclude this work and state some open problems. Background on information theory and private computations is given in the appendices.

Chapter 2

History

In this chapter we give a short description of works dealing with secret sharing schemes and key distribution schemes.

2.1 Secret Sharing Schemes

Threshold secret sharing schemes were first introduced in 1979 by Shamir [101] and Blakley [22]. The size of the pieces in Shamir's scheme is equal to the size of the secrets. Other threshold schemes were described in [69, 72, 79]. Their properties were studied in [15, 41, 50, 69, 72, 79, 109]). They were used in many applications, e.g., Byzantine agreement in [93], private computations (e.g. [13, 39, 59]), threshold cryptology (e.g. [49]), and zero knowledge (e.g [47]). Secret sharing in which the secret can be reconstructed by the human visual system was presented in [85]. Reducing the size of the pieces by compromising on computational security was considered in [73].

Ito, Saito, and Nishizeki [62] generalized the notion of secret sharing to general access structures. They show that every monotone access structure has a secret sharing scheme that realizes the access structure. Benaloh and Leichter [16] describe more efficient schemes which are based on monotone formulas that describe the access structure. Many other schemes were presented. Simmons [103, 104], and Simmons, Jackson and Martin [105] describe schemes based on geometric configurations. Brickell and Davenport describe schemes based on matroid representation [34, 35] (see also Karchmer and Wigderson [68] and Bertilsson and Ingemarsson [19]). These are the schemes we call linear since in these schemes the pieces are linear combinations of random strings. Jackson and Martin [64] proved that linear schemes are equivalent to the geometric schemes. Schemes based on the decomposition technique were discussed in [16, 27, 30, 36, 107, 108]. The survey of Stinson [106] contains a description of most of these scheme. Naor and Wool [86] suggest the use of these generalized schemes to enable access to a secure database. Other applications in which secret sharing are used are described in the survey of Simmons [104].

We briefly describe some results concerning lower bounds on the size of the pieces. The first observation is that in every secret sharing scheme realizing any access structure the size of the piece of every party is at least the size of the secret [69, 38] (actually, this result follows from Shanon's classical result [102] about perfect encryption systems). Benaloh and Leichter [16] describe an explicit access structure Γ, that cannot achieve this lower bound. The description of Γ appears in Fig. 2.1, where the edges represents the minimal authorized sets.

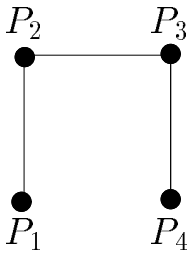

Figure 2.1: Description of the access structure Γ.
איור 2.1: תאור של מבנה הגישה Γ.

Brickell [34] defined the notion of *ideal* secret sharing schemes. These are secret sharing schemes that achieve the lower bound, that is schemes in which the size of the piece of every party equals the size of the secret. Brickell and Davenport [35] "almost" characterized ideal schemes in terms of matroids. Access structures, which have ideal schemes over any finite domain of secrets are called *universally ideal*. In my M.Sc. thesis [6, 8], we defined this notion and gave an exact characterization of these access structures. Ideal schemes were also considered in [36, 63, 75, 100].

Capocelli, De Santis, Gargano, and Vaccaro [38], using the entropy function, prove better lower bounds on the size of the pieces for the access structure Γ (see Fig. 2.1). They prove that in every secret sharing scheme realizing Γ there exists a party that receives pieces of size at least 1.5 times the size of the secret (for every domain of secrets). Better lower bounds (for other access structures) were presented in [27, 28, 53]. The best lower bound is due to Csirmaz [44, 43]. His proof gives, for every n, an access structure with n parties for which the sum of the sizes of the pieces in every secret sharing scheme is $\Omega(n^2/\log n)$ times the size of the secret (for every finite set of possible secrets).

An important lower bound is given by Kilian and Nisan [71] for threshold schemes. They prove that in every t-out-of-n threshold secret sharing scheme there exists a party with pieces of size at least $\log(n-t+2)$ (for $2 \leq t \leq n-1$). Unlike the previous results, this lower bound does not increase as the secrets become longer. This lower bound explains why in Shamir's scheme [101] the cardinality of the domain of pieces has to be larger than n.

A subject related to sharing a secret with respect to a family of functions is multi-secret sharing schemes. That is, there are many secrets that should be shared simultaneously. There are a few variations of this question. In [65, 66] each set of cardinality at least t should be able to reconstruct some of the secrets (but not all). In non-perfect secret sharing schemes (considered in [23, 59, 69, 79]) coalitions of at most ℓ parties know nothing on the secrets, sets of at least t party can reconstruct all secrets (where ℓ and t are parameters such that $\ell < t$). Sets of size between ℓ and t know nothing about each secret, but are allowed to have some information on the dependence between the different secrets. In [26, 31, 66] each set B in an access structure is able to reconstruct any of the distributed secrets. The security requirement considered in [31, 66, 26] is somewhat weak: after revealing one of the secrets, limited information about other secrets may be leaked. Similar scenarios to sharing with respect to a family of functions, in which sharing is viewed as a form of encryption and the security is computationally bounded, have been considered in [1, 5, 98].

Public channels have been used in secret sharing (in addition to private channels) in dynamic sharing of secrets. These are schemes where the dealer enables parties to reconstruct different secrets in different time instants (e.g. [104, 21, 25]). A different scenario in which a public broadcast channel is used (in addition to private channels) is to protect against Byzantine parties [14]. Unlike our scenario, in that work the broadcast channel is heard by *all* parties.

2.2 Key Generation and Key Distribution Schemes

The problem of key generation is one of the basic problems in cryptology. Therefore, it received a lot of attention. The idea of key exchange was introduced by Merkle [80] and in the pioneering work of Diffie and Hellman [51] on public key cryptography. Diffie and Hellman [51] suggested a key generation protocol for conferences of size two. The Diffie Hellman protocol is as follows: Let p be a prime number, and let α be a primitive element in the field GF(p). Party P_i (respectively P_j) chooses a random number $r_i \in \mathrm{GF}(p)$ (respectively r_j) and sends the message $m_i = \alpha^{r_i}$ (respectively $m_j = \alpha^{r_j}$). The joint key of P_i and P_j is $\alpha^{r_i \cdot r_j}$, which P_i easily computes from m_j and r_i using the equality $m_j^{r_i} = \alpha^{r_i \cdot r_j}$. A short list of papers that deal with this subject in the computational model are [78, 113, 12]. Practical key distribution systems are Kerboros [89] and KryptoKnight [82, 20]. A recent survey on key distribution is [99].

Blom [24] was the first to consider information theoretic key distribution schemes. He presented an efficient non-communicating scheme, based on MDS codes, for conference of size 2 and coalitions of size b. Matsumoto and Imai [76] suggested the use of symmetric linear functions for schemes for larger conferences. The most efficient scheme for larger conferences was presented by Blundo et. al. [29]. They generalize Blom's construction and present schemes based on symmetric multinomials. Their multinomials have g variables and degree b in each variable. The size of the pieces in their scheme is $\binom{g+b-1}{g-1}$ times the size of

the keys. For large values of g and b, this expression is quite large. However, using entropy arguments, Blundo et. al. [29] prove a tight $\binom{g+b-1}{g-1}$ times size of keys lower bound on the size of pieces. Therefore, their scheme is space-optimal. The number of random bits required in these schemes was analyzed in [32]. Other non-communicating schemes in this setting were presented in [81, 90, 61, 92].

Blundo et. al. [29] also presented one-time secure communicating scheme in which the size of the pieces is $g + b - 1$ times the size of the keys. (The fact that this scheme is only one-time secure was not mentioned in [29]). One time secure communicating schemes based on a random deal of cards are discussed in [55, 56, 57, 58].

Chapter 3

Definitions

In this chapter we define our model, secret sharing schemes, and key distribution schemes.

3.1 The Model

We consider a system with n parties denoted by $\{P_1, P_2, \ldots, P_n\}$. In addition to the parties, there is a dealer in the system, who has an input x. A *scheme* is a probabilistic mapping, which the dealer applies to the input, and generates n pieces of information. For example, x is a secret whom the dealer wants to share. Formally,

Definition 3.1 [Scheme]: *Let X be a set of inputs, R be a set of random inputs, and $\mu : R \to [0,1]$ be a probability distribution on the random inputs R. A* scheme Π *is a mapping $\Pi : X \times R \to S_1 \times S_2 \times \ldots \times S_n$ from the cross product of the inputs and the random inputs to a set of n-tuples, called the* pieces *(sometimes referred to as* shares*). The coordinate i of the n-tuple $\Pi(x,r)$ is called the piece of P_i, and is denoted by $\Pi_i(x,r)$, and S_i is the domain of pieces of P_i. We refer to Π as the* dealer*, which has an input x and generates the pieces. Given an input $x \in X$, the dealer chooses a random input r according to the distribution μ, and generates the vector of pieces $\Pi(x,r)$. The dealer gives the i-th piece to P_i in a private way, i.e. the other parties have no information on the piece of P_i other than the information inferred from their pieces.*

In the scenario we consider, the dealer is only active during the initialization of the system. After the initialization stage, the parties can communicate. We next define the two models of communication we consider.

Definition 3.2 [Communication Models]: *The parties communicate via a complete synchronous network of point-to-point communication channels. We consider two models of the security of the channels:*

- *The* secure private channels *model in which the parties communicate via secure communication channels with no eavesdropping.*

- *The* insecure channels *model in which the "curious" parties can overhear all communications exchanged between all parties in the system. So, from the point of view the "curious" parties the channels are* public. *On the other hand, "good" parties hear only messages sent to them. (In particular, from the point of view of the "good guys", the channels do not carry any of the potential advantages of a broadcast channel.)*

In both cases the channels are reliable, that is, the messages are received as they are sent and no party can change these messages, or impersonate other parties.

A subset of the parties can communicate in order to compute a function of their pieces (e.g. the shared secret, a common key). We now define how they evaluate this function.

Definition 3.3 [Protocol]: *A subset of the parties execute a protocol to evaluate a function f. At the beginning of an execution, each P_i has an input x_i and a random input r_i. The parties exchange messages, as prescribed by the protocol. In each round every party sends a message to every other party. Each message sent by a party is determined by its input, its random input, the messages it received so far, and the identity of the receiver. We say that a* protocol computes the function *f if each party that took part in the protocol can evaluate the correct value of f from its input and the communication it heard. A protocol is* non-interactive *if the messages sent by each party depend only on his input and his random input (and not on messages received during the execution of the protocol). That is, the protocol has only one-round of communication. If the protocol has more than one round then it is called* interactive. *In this case we require that the protocol terminates after a finite number of rounds (that is, we do not allow infinite runs).*

In this work we consider only honest parties (no Byzantine parties). That is, the parties follow their protocols. However, they are curious and after the protocol has ended some of them can collude and try to gain some partial information (e.g. on the pieces of parties or on the input of the dealer). We consider the information theoretic model in which the "curious" parties, which have unlimited power, are not allowed to gain any information, as defined in the next definition:

Definition 3.4 [No Information]: *Let B be a (curious) coalition (set of parties). The* view *of B, denoted by $VIEW_B$, after an execution of a protocol is all the information it has, i.e. the pieces of the parties in the coalition, their local random inputs, and the messages they heard. In the secure private channels these are the messages that are sent to parties in the coalition. In the insecure channels model these are the messages exchanged by* all *parties over the communication channels (here the insecurity of the communication is manifested).*

The coalition B has no information *on a random variable* X *if for every two possible values* x_1, x_2 *of* X*, and every value of* $VIEW_B$*:*

$$\Pr[\ VIEW_B \mid X = x_1\ ] \ = \ \Pr[\ VIEW_B \mid X = x_2\ ]\ ,$$

where the probability is taken over the random inputs of the dealer, and the random inputs of the parties outside the coalition. Notice that we do not make any assumptions on the distribution of X*.*

We will be interested in efficient schemes. More precisely we shall consider space efficiency of the schemes. That is, we consider the pieces of the parties as binary strings, and the (space) *efficiency* of a scheme is the size of the strings that represent the pieces of the parties.

3.2 Secret Sharing Schemes

We define (generalized) secret sharing scheme. The definition does not specify the communication model during the reconstruction of the secret. We require that after the reconstruction the secret remains unknown to parties not participating in the reconstruction. Therefore, implicitly the definition assumes secure private channels between the parties. (In latter chapters we will define other variations of this schemes.) We first define the notion of an access structure. This is a collection of sets of parties which should be able to reconstruct the secret.

Definition 3.5 [Access Structure]: *Let* $\{P_1, \ldots, P_n\}$ *be the set of parties. A collection* $\mathcal{A} \subseteq 2^{\{P_1,\ldots,P_n\}}$ *is monotone if* $B \in \mathcal{A}$ *and* $B \subseteq C$ *implies* $C \in \mathcal{A}$*. An* access structure *is a monotone collection* $\mathcal{A}$ *of non-empty subsets of* $\{P_1, \ldots, P_n\}$ *(that is,* $\mathcal{A} \subseteq 2^{\{P_1,\ldots,P_n\}} \setminus \{\emptyset\}$*). The sets in* $\mathcal{A}$ *are called the* authorized sets*, and the sets not in* $\mathcal{A}$ *are called the* unauthorized sets*.*

A secret sharing scheme realizing an access structure $\mathcal{A}$ is a scheme, in which the dealer has a secret input $s \in S$, and generates private pieces for the parties. Any authorized set of parties can reconstruct the secret from its pieces, and any unauthorized set not in $\mathcal{A}$ cannot reveal any partial information about the secret. Formally,

Definition 3.6 [Secret Sharing]: *Let* S *be a finite domain of secrets. A* secret-sharing scheme *realizing an access structure* $\mathcal{A}$ *is a scheme in which the input of the dealer is a secret* $s \in S$ *such that the following two requirements hold:*

Reconstruction requirement *The secret* s *can be reconstructed by any authorized set. That is, for any set* $G \in \mathcal{A}$ $(G = \{i_1, \ldots, i_{|G|}\})$*, there exists a reconstruction function* $h_G : S_{i_1} \times \ldots \times S_{i_{|G|}} \rightarrow S$ *such that for every secret* s*, and every random input* r*,*

$$\textit{if}\ \ \Pi(s, r) \ = \ \langle s_1, s_2, \ldots, s_n \rangle \ \ \textit{then}\ \ h_G(s_{i_1}, \ldots, s_{i_{|G|}}) \ = \ s.$$

Security requirement *Every unauthorized set of parties cannot reveal any partial information about the secret as defined in Definition 3.4. We state this condition explicitly: for any set $B \notin \mathcal{A}$, for every two secrets $a_1, a_2 \in S$, and for every vector of possible pieces $\{s_i\}_{i \in B}$:*

$$\Pr[\bigwedge_{P_i \in B} \Pi_i(a_1, r) = s_i] \quad = \quad \Pr[\bigwedge_{P_i \in B} \Pi_i(a_2, r) = s_i] \ .$$

Where the probabilities are taken over the random input of the dealer.

A secret sharing scheme realizing the access structure $\{G : |G| \geq t\}$ is called a *t-out-of-n threshold secret sharing scheme*. These are the schemes which were defined by Blakley [22] and Shamir [101] and have received most attention.

3.3 Key Distribution Schemes

In this section, we present formal definition of key distribution schemes. In these schemes every set G of size g can generate a common key. This key should be known to each party of G. On the other hand, every coalition of at most b parties should not have any information about this key. We consider two variants:

- a *non-communicating* scheme in which, after the initial distribution of the pieces, the generation of the keys is done without any communication. That is, every party in the set reconstructs the key from his piece.
- A *communicating* scheme in which the parties are allowed to communicate during the generation of the key.

We start with non-communicating schemes:

Definition 3.7 [Non-Communicating Key Distribution Schemes]: *Let g and b be positive integers such that $g + b \leq n$, the set K be a set of keys, $\mathcal{P}$ be an a-priori probability distribution on K, and U_i be the domain of pieces for party P_i. A* non communicating (g, b) key distribution scheme *is a scheme $\mathcal{U} : R \rightarrow U_1 \times U_2 \times \ldots \times U_n$ such that:*

Reconstruction requirement *Every conference (set) G of cardinality g has a common key; this key is determined by the random input of the dealer and is denoted $k_G(r)$. It is required that this key is distributed according to the a-priori probability distribution $\mathcal{P}$; that is for every $k \in K$*

$$\Pr\ [\ k_G(r) = k\ ] \ = \ \mathcal{P}(k) \ ,$$

where the probability is taken over the random input of the dealer r. Each member of G can deterministically reconstruct G's key from his piece without any communication *with the dealer or other parties.*

Non-communicating security requirement *Every ("curious") coalition B of cardinality b does not gain any information from its pieces about a key of a disjoint conference G ($G \cap B = \emptyset$). That is, for every vector of pieces $\langle u_1, \ldots, u_n \rangle$, and every two keys $k_1, k_2 \in K$:*

$$\Pr\left[\bigwedge_{P_j \in B} (\mathcal{U}_j(r) = u_j) \mid \ k_G(r) = k_1 \right] = \Pr\left[\bigwedge_{P_j \in B} (\mathcal{U}_j(r) = u_j) \mid \ k_G(r) = k_2 \right] .$$

The probability is taken over r – the random input of the dealer.

It is *not* guaranteed that keys of different conferences are *independent* random variables. The security requirement does imply some independence between the keys. For example, it is true that every $b+1$ keys are independent. Otherwise, there are b parties knowing the first b keys, and therefore gain some information on the $(b+1)$ key of a disjoint conference. Claim 6.2 gives another scenario in which this independence is guaranteed. In the rest of this work we assume that the a-priori probability of each key is positive. That is, for every key $k \in K$ it holds that $\mathcal{P}(k) > 0$.

Next we consider communicating schemes. In this case the communication in the generation of one key might leak information on keys of other conferences. Therefore, unlike non-communicating schemes, we consider two security requirements:

- Restricted schemes in which the secrecy of the generated keys is guaranteed only if a restricted number of sets generate a key.
- Unrestricted schemes in which the secrecy of the generated keys is guaranteed even if an unrestricted number of conferences (possibly all) generate a key (however, each conference is allowed to generate only one key).

Before going any further, we remark that the notion of key distribution schemes restricted to a limited number of conferences is meaningful only with respect to communicating schemes. For non-communicating schemes, the generation of a conference key does not add any information to any party (either in the conference or not). Therefore a one-time secure non-communicating scheme would also be secure in the unrestricted sense, and no saving can be expected. On the other hand, the communication in communicating schemes is heard by all parties (not only conference members), and could reduce the uncertainty of the remaining pieces. It is possible that after some communications took place, no uncertainty is left, and the pieces become useless for additional conferences. This implies that the amount of initial secrecy in restricted communicating schemes can be smaller than that of unrestricted schemes. We first define unrestricted communicating schemes:

Definition 3.8 [Unrestricted Communicating Key Distribution Schemes]: *An* unrestricted communicating (g, b) key distribution scheme *with n parties and domain of keys K is a scheme $\mathcal{U} : R \rightarrow U_1 \times U_2 \times \ldots \times U_n$ in which every conference (set) G of cardinality*

g can generate a common key after communicating over insecure channels. We denote the communication which is exchange while G generates the key by $C_G(r, \vec{r}_G)$.[1]

Reconstruction requirement *At the end of the conversation, each member of G can deterministically reconstruct a key from the conversation and his piece. The key that every member of G reconstructs is the same, and is denoted by* $k_G(r, \vec{r}_G)$*, where r is the random input of the server, and* $\vec{r}_G$ *is the vector of random inputs of the parties in G. It is required that this key is distributed according to the a-priori probability distribution* $\mathcal{P}$*; that is for every* $k \in K$

$$\Pr\left[\ k_G(r, \vec{r}_{G_0}) = k\ \right] = \mathcal{P}(k)\ ,$$

where the probability is taken over r, the random input of the server, and $\vec{r}_G$ *the random inputs of the parties of G.*

Unrestricted security requirement *Any ("curious") coalition B of cardinality b, having their pieces and random inputs, and knowing the conversations of* all *possible conferences, does not gain any information on the key of any disjoint subset* G_0 *(i.e.* $G_0 \cap B = \emptyset$*). That is, the view of the coalition B is*

$$VIEW_B = \vec{r}_B \wedge \bigwedge_{P_j \in B} u_j \wedge \bigwedge_{|G|=g} C_G\ ,$$

where $\langle u_j \rangle_{P_j \in B}$ *is any vector of pieces;* $\vec{r}_B$ *is any set of random inputs of the coalition members; and* $C_1, \ldots, C_{\binom{n}{g}}$ *are any possible conversations of all sets of cardinality g. We require that for every two keys* $k_1, k_2 \in K$*:*

$$\Pr\left[\ VIEW_B \mid k_{G_0}(r, \vec{r}_{G_0}) = k_1\ \right] = \Pr\left[\ VIEW_B \mid k_{G_0}(r, \vec{r}_{G_0}) = k_2\ \right].$$

The probability is taken over r – the random input of the server, and over $\vec{r}$ *– the random inputs of all the parties for all conferences;* $\vec{r}_B$ *is the restriction of* $\vec{r}$ *to the coalition B.*

We continue with the definition of τ-restricted key distribution scheme. These are communicating key distribution scheme in which the secrecy of the key is guaranteed only if at most τ conferences generated a key.

Definition 3.9 [Restricted Communicating Key Distribution Schemes]: *A* τ-restricted communicating (g, b) key distribution scheme *is a communicating* (g, b)*-scheme in which the security property is replaced by the following one:*

τ-restricted security property *Any ("curious") coalition B of cardinality b, having their pieces and random inputs, and knowing the conversations of any* τ *conferences*

[1] It is a function of the g pieces and the local random inputs, but since the pieces are determine by the random input of the dealer then we shall also consider it as a function of the dealer's random input and the local random inputs.

$G_0, \ldots, G_{\tau-1}$, does not gain any information on the key of the disjoint subset G_0 (i.e. $G_0 \cap B = \emptyset$). That is, the view of the coalition B is

$$VIEW_B = \vec{r}_B \wedge \bigwedge_{P_j \in B} u_j \wedge \bigwedge_{0 \le j \le \tau-1} C_{G_j} \ ,$$

where $\langle u_j \rangle_{P_j \in B}$ is any vector of pieces; $\vec{r}_B$ is any set of random inputs of the coalition members; and $C_0, C_1, \ldots, C_{C_{\tau-1}}$ are any possible conversations of any combination of τ sets $G_0, \ldots, G_{\tau-1}$ of cardinality g. We require that for every two keys $k_1, k_2 \in K$:

$$\Pr\left[\ VIEW_B \mid k_{G_0}(r, \vec{r}_{G_0}) = k_1\ \right] \ = \ \Pr\left[\ VIEW_B \mid k_{G_0}(r, \vec{r}_{G_0}) = k_2\ \right] .$$

The probability is taken over r – the random input of the server, and over $\vec{r}$ – the random inputs of all the parties for all conferences; $\vec{r}_B$ is the restriction of $\vec{r}$ to the coalition B. We denote 1-restricted scheme by one-time scheme.

Chapter 4

Linear Secret Sharing Schemes and Monotone Span Programs

In this chapter we consider linear secret sharing schemes. These are schemes in which the reconstruction functions are linear. That is, every piece is a vector over some finite field, and every set in the access structure reconstructs the secret using a linear combination of the coordinates of its pieces. Most secret sharing schemes that were proposed are linear, e.g. [16, 19, 22, 27, 30, 35, 36, 52, 62, 68, 72, 101, 103, 104, 105, 107, 108]. The existence of an efficient linear schemes for a specific access structure is equivalent to the existence of a small monotone span programs – a computation model presented in [68]. Span programs are a linear algebraic model of computation. Monotone span programs provide an easier model for proving lower bounds for linear secret sharing schemes. Lower bounds for monotone span programs also imply lower bounds for monotone formulae, monotone symmetric branching programs, and for monotone contact schemes. In this chapter we define linear secret sharing schemes and monotone span programs, and prove the equivalence between them. In Chapter 5 we prove lower bounds on the size of span programs.

4.1 Linear Secret Sharing Schemes

We first define linear secret sharing schemes. These are a special class of secret sharing schemes in which the reconstruction functions are linear.

Definition 4.1 [Linear secret sharing schemes]: *Let $\mathcal{K}$ be a finite field, and Π be a secret sharing scheme with domain of secrets $S \subseteq \mathcal{K}$ realizing an access structure $\mathcal{A}$. We say that Π is a* linear secret sharing scheme *over $\mathcal{K}$ if:*

1. *The piece of each party is a vector over $\mathcal{K}$. That is, for every i there exists a constant d_i such that the piece of P_i is taken from $\mathcal{K}^{d_i}$. We denote by $\Pi_{i,j}(s,r)$ the j-th coordinate in the piece of P_i (where $s \in S$ is a secret and $r \in R$ is the dealer's random input).*

2. *For every authorized set, the reconstruction function of the secret from the pieces is linear. That is, for every $G \in \mathcal{A}$ there exist constants $\{\alpha_{i,j} : P_i \in G, 1 \leq j \leq d_i\}$, such that for every secret $s \in S$ and every choice of random inputs $r \in R$,*

$$s = \sum_{P_i \in G} \sum_{1 \leq j \leq d_i} \alpha_{i,j} \cdot \Pi_{i,j}(s, r)$$

where the constants and the arithmetic are over the field $\mathcal{K}$.

The total size *of the pieces in the scheme is defined as $d \triangleq \sum_{i=1}^{n} d_i$.*

To show that the definition of linear schemes is natural, we show that it is equivalent to a few alternative definitions. In Section 4.4 we show that it equivalent to the existence of monotone span programs. Jackson and Martin [64] proved that linear schemes are equivalent to geometric schemes introduced in [103, 104, 105]. Furthermore, we state an alternative definition, and show that this definition is equivalent to the original definition. While in the original definition the reconstruction of the secret is linear, in the alternative definition the generation of the pieces by the dealer is linear. Formally,

Definition 4.2 [Alternative Definition]: *A secret sharing scheme is linear (linear generation of pieces) if:*

1. *The piece of each party is a vector over $\mathcal{K}$.*
2. *During the generation of the pieces, the dealer chooses independent random variables, denoted $r_1, \ldots, r_\ell$, each one distributed uniformly over $\mathcal{K}$. Each coordinate of the piece of every party is a linear combination of $r_1, \ldots, r_\ell$ and the secret s.*

We show that the latter definition implies the original definition:

Claim 4.3: *Every scheme that is linear according to Definition 4.2 is linear according to Definition 4.1.*

Proof: Consider an authorized set $G \in \mathcal{A}$. Each coordinate of the pieces of the parties in G is a linear combination of the random inputs of the dealer and the secret. We can write these combinations as a system of linear equations in which the unknowns are the secret and the random inputs of the dealer. Since G can reconstruct the secret there is only one secret $s_0 \in S$ that is consistent with this system (by consistent we mean that there exists a solution to the system in which the secret equals s_0). However, $S \subseteq \mathcal{K}$ and, presumably, there can be an element $s_1 \in \mathcal{K} \setminus S$ that is consistent with this system. In this case, for every $\alpha \in \mathcal{K}$ the element $\alpha s_0 + (1 - \alpha)s_1$ is consistent with the system. But $\alpha s_0 + (1 - \alpha)s_1$ ranges over all the field $\mathcal{K}$ and all secrets would have been consistent with the system. Therefore, s_0 is the only value in $\mathcal{K}$ for the secret that is consistent with the system. Thus, the equation $s = s_0$ is a linear combination of the equations in the linear system, and the secret is a linear combination of the coordinates of the pieces in G. □

In Claim 4.9 and Claim 4.7 we prove the converse of this claim; that is if there exists a linear scheme according to Definition 4.1 then there exists a linear scheme according to Definition 4.2 whose size is the same.

4.2 Examples of a Linear Secret Sharing Schemes

In this section we describe two known secret sharing scheme, and show that they are linear.

Example 4.4 [Shamir t-out-of-n threshold scheme [101]]: Let q be the size of the domain of secrets, where q is a prime-power that is bigger than n (the number of parties in the access structure). Let $s \in \mathrm{GF}(q)$ be the secret. The dealer chooses independently with uniform distribution $t-1$ random elements in $\mathrm{GF}(q)$, which are denoted by $r_1, \ldots, r_{t-1}$. These elements and the secret s define a polynomial

$$p(x) \triangleq r_{t-1}x^{t-1} + r_{t-2}x^{t-2} + \ldots + r_1 x + s \ .$$

Observe that $p(0) = s$. The dealer gives the piece $p(i)$ to party P_i. This piece is a linear combination of the random inputs and the secret. Now each set of cardinality at least t can reconstruct $p(x)$ by interpolation. That is, the set $\{P_{i_1}, \ldots, P_{i_t}\}$ holding the pieces $\{s_{i_1}, \ldots, s_{i_t}\}$, can compute the polynomial $p(x)$ since

$$p(x) = \sum_{j=1}^{t} s_{i_j} \prod_{d \neq j} \frac{i_d - x}{i_d - i_j}.$$

The secret is reconstructed by substituting 0 for x in this polynomial. That is, the secret is a linear combination of the pieces $\{s_{i_1}, \ldots, s_{i_t}\}$, where the coefficient of the piece s_{i_j} of party P_{i_j} is $\prod_{d \neq j} \frac{i_d}{i_d - i_j}$. Therefore, Shamir's scheme is linear (according to both definitions).

Example 4.5 [Scheme for Monotone Symmetric Branching Programs [17]]: We next describe an interesting linear scheme which was originally presented by Benaloh and Rudich [17] (see also [68]). We first describe a computation model called monotone symmetric branching programs, also known as monotone undirected contact schemes and switching networks (for more details on this subject the reader can refer to [68, 97]). Let $H = (V, E)$ be an *undirected graph*, $\rho : E \to \{x_1, \ldots, x_n\}$ be a labeling of the edges by variables, and v_0 and v_1 be two special vertices in the graph. A monotone symmetric branching programs is defined as $\langle H, \rho, v_0, v_1 \rangle$. This program computes the following Boolean function: Given an assignment $\delta \in \{0,1\}^n$ define H_δ to be the subgraph whose vertices are the same vertices as H, and the edges are all the edges whose labels are variables x_i such that the i-th bit of δ equals 1. The program accepts δ if in H_δ there exists a path from v_0 to v_1. Similarly, this program defines the access structure whose parties are $\{P_1, \ldots, P_n\}$ and its authorized sets are all the sets whose characteristic vectors[1] are accepted by the contact scheme.

We show that every access structure defined by a monotone symmetric branching program with d edges has a linear scheme in which the total of the pieces size is d. Let $\mathcal{K}$ be the domain of secrets (where $\mathcal{K}$ is any finite field), and $\left\{r_2, \ldots, r_{|V|-1}\right\} \in \mathcal{K}^{|V|-1}$ be random

[1]Given a set $G \in \{P_1, \ldots, P_n\}$ its characteristic vector $\delta_G \in \{0,1\}^n$ is the vector in which the i-th coordinate equals 1 if and only if $P_i \in G$.

inputs of the dealer. Define $r_0 \triangleq 0$ and $r_1 \triangleq s$. For every edge $(v_i, v_j) \in E$ assign the value $r_i - r_j$. The piece of P_i are all the values assigned to edges labeled by x_i. We can represent this scheme by an $|E| \times |V|$ matrix in which each row is indexed by an edge and every column is labeled by a vertex. The row indexed by the edge (v_i, v_j) (where $i \leq j$) is 1 in the v_i coordinate, -1 in the v_j coordinate and is 0 otherwise. Using the notation of the following section, this matrix is a monotone span program for the function.

We claim that this scheme realizes the access structure defined by the monotone symmetric branching program. For every simple path which starts at v_0, and ends at v_1 , it is possible to assign ± 1 weights to the values assigned to edges along the path, such that the weighted sum is equal to the secret s. Therefore, every authorized set can reconstruct the secret. On the other hand, since every unauthorized set does not contain at least one cut in the graph[2], the unauthorized set has no information on the secret.

4.3 Span Programs

We state the definition of the model of span programs from [68]. A span program for a Boolean function is presented as a matrix over some field with rows labeled by literals of the variables. The span program accepts an assignment if and only if the all-ones row is a linear combination of the rows whose labels are consistent with the assignment.

Definition 4.6 [Span Programs]: *Let $\mathcal{K}$ be a field, and $\{x_1, \ldots, x_n\}$ be a set of variables. A* span program *over $\mathcal{K}$ is a labeled matrix $\hat{M}(M, \rho)$ where M is a matrix over $\mathcal{K}$, and ρ is a labeling of the rows of M by literals from $\{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}$ (every row is labeled by one literal).*

A span program accepts or rejects an input by the following criterion. For every input sequence $\delta \in \{0,1\}^n$ define the submatrix M_δ of M consisting of those rows whose labels are set to 1 by the input δ, i.e., either rows labeled by some x_i such that $\delta_i = 1$ or rows labeled by some $\bar{x}_i$ such that $\delta_i = 0$. The span program $\hat{M}$ accepts *δ if and only if $\vec{1} \in \mathrm{span}(M_\delta)$, i.e., some linear combination of the rows of M_δ gives the all-one vector $\vec{1}$. (The row vector $\vec{1}$ has the value 1 in each coordinate.) A span program* computes *a Boolean function f if it accepts exactly those inputs δ where $f(\delta) = 1$.*

A span program is called monotone *if the labels of the rows are only the positive literals $\{x_1, \ldots, x_n\}$. Monotone span programs compute monotone functions.*

The size *of $\hat{M}$ is the number of rows in M. We denote by $\mathrm{SP}_{\mathcal{K}}(f)$ (respectively $\mathrm{mSP}_{\mathcal{K}}(f)$) the size of a smallest span program (respectively monotone span program) over $\mathcal{K}$ that computes f.*

The vector $\vec{1}$ in the definition above can be replaced by any fixed nonzero vector (as will sometimes be convenient) via a change of basis. This vector is called the objective vector.

[2] A cut in a connected graph is a set of edges whose removal from the graph results in a non-connected graph.

The class of functions with polynomial size span programs is equivalent to the class of functions with polynomial size counting branching programs [37, 68]. Span program size is a lower bound on the size of symmetric branching programs [68] (see Example 4.5). Lower bounds for span programs also imply lower bounds for formula size. It is known that every function with a polynomial size span program is in NC; this follows from the fact that linear algebra is in NC (see [18, 37, 68, 83]).

Monotone span programs have only positive literals (non-negated variables) as labels of the rows. They compute only monotone functions, even though the computation uses non-monotone linear algebraic operations. In fact, Babai et. al. [4] showed a function whose monotone circuit complexity is super-polynomial although it has a linear size monotone span program. Thus, lower bounds on monotone circuit size do not imply lower bounds on the size of monotone functions. However, the reduction in [68] from symmetric branching programs to span programs, described in Example 4.5, preserves monotonicity, and thus lower bounds for monotone span programs imply lower bounds for monotone symmetric branching programs and for monotone formula size.

4.4 Equivalence of Span Programs and Linear Secret Sharing Schemes

In this section we show that linear secret sharing schemes and monotone span programs are equivalent. To make this statement formal we need some notations. Given a set $G \in \{P_1, \ldots, P_n\}$ let $\delta_G \in \{0,1\}^n$ be its characteristic vector, that is the i-th coordinate equals 1 if and only if $P_i \in G$. Define the function $f_{\mathcal{A}} : \{0,1\}^n \to \{0,1\}$ as:

$$G \in \mathcal{A} \qquad \text{if and only if} \qquad f_{\mathcal{A}}(\delta_G) = 1 \ .$$

In the following two claims we quote the proofs that the size of a smallest monotone span program computing $f_{\mathcal{A}}$ is equivalent to the size a smallest linear secret sharing scheme realizing $\mathcal{A}$. The first claim was proved by Brickell [34], Brickell and Davenport [35], and Karchmer and Wigderson [68].

Claim 4.7: [34, 35, 68] *Assume there exists a monotone span program, of size d, over a finite field $\mathcal{K}$ computing the function $f_{\mathcal{A}}$. Then there is a linear secret sharing scheme over $\mathcal{K}$ realizing the access structure $\mathcal{A}$ in which the total size of the pieces is d.*

Proof: Let $\hat{M}$ be a monotone span program with ℓ columns. We construct a linear secret sharing scheme in which the dealer chooses at random a vector $\vec{r} = \langle r_1, \ldots, r_\ell \rangle$ from $\mathcal{K}^\ell$ such that the sum of its coordinates is equal to the secret, i.e. $\vec{1} \cdot \vec{r} = \sum_{i=1}^{\ell} r_i = s$. Consider the vector $M\vec{r}$, and label each of its coordinates according to the labeling of the corresponding row in $\hat{M}$. The piece of P_i are all the coordinates of $M\vec{r}$ that are labeled by x_i. We will

show that this scheme is a secret sharing scheme realizing $\mathcal{A}$. That is, every set $G \in \mathcal{A}$ can reconstruct the secret, while every set $B \notin \mathcal{A}$ has no information on the secret.

Let G be an authorized set in $\mathcal{A}$. Since $\hat{M}$ computes the function $f_{\mathcal{A}}$, there exists a combination of rows indexed by variables in G that equals to the all one vector. That is, in $\hat{M}$ there exists rows, denoted $\vec{M}_1, \ldots, \vec{M}_d$, whose labels are from G, and there exist constants $\beta_1, \ldots, \beta_d$ such that $\sum_{i=1}^{d} \beta_i \vec{M}_i = \vec{1}$. Furthermore, by our construction the parties in G hold the values $\vec{M}_1 \cdot \vec{r}, \ldots, \vec{M}_d \cdot \vec{r}$. The "weighted sum" of these values equals to the secret:

$$\sum_{i=1}^{d} \beta_i \left(\vec{M}_i \cdot \vec{r} \right) = \left(\sum_{i=1}^{d} \beta_i \vec{M}_i \right) \cdot \vec{r} = \vec{1} \cdot \vec{r} = s \ .$$

That is, every authorized set can reconstruct the secret, by applying a linear function to the coordinates of their pieces.

Now, let B be an unauthorized set (a set not in $\mathcal{A}$). We have to prove that the parties in B have no information about the secret. That is, the number of possible random strings that are consistent with pieces of B given a secret s_0 is equal to the number of possible random strings that are consistent with the pieces of B given a secret s_1. This will be done by showing a one to one mapping from the random strings that are consistent with the pieces and the secret s_0 to the random strings that are consistent with the pieces and the secret s_1. Furthermore, this mapping has an inverse, so the cardinality of the two sets of random inputs is equal.

To exhibit such a mapping, we need the following proposition from linear algebra. It is a special case of the fact that a linear space V is a subspace of a linear space N if and only if the null space of V *contains* the null space of N.

Proposition 4.8: *A vector $\vec{v}$ is independent of a set of vectors represented by a matrix N if and only if there exists a vector $\vec{w}$ such that $N\vec{w} = \vec{0}$ while $\vec{v} \cdot \vec{w} \neq 0$.*

Since $\vec{1}$ is independent of M_{δ_B}, there exists a vector $\vec{r'}$ such that $M_{\delta_B}\vec{r'} = \vec{0}$ while $\vec{1} \cdot \vec{r'} \neq 0$. Now, let $\vec{c}$ be a possible vector of pieces of the set B, and $\vec{r}$ be a random vector that the dealer used to generate the vector $\vec{c}$. Finally, let s_0 its corresponding secret. That is, $M_{\delta_B}\vec{r} = \vec{c}$ and $\vec{1} \cdot \vec{r} = s_0$. Define the following mapping from the random vectors consistent with s_0 to the random vectors consistent with s_1:

$$\phi(\vec{r}) \triangleq \vec{r} + \alpha\vec{r'} \ ,$$

where $\alpha \triangleq (s_1 - s_0)/(\vec{1} \cdot \vec{r'})$. Since $\vec{r}$ generates the pieces $\vec{c}$, the vector $\phi(\vec{r})$ generates them as well:

$$M_{\delta_B}(\vec{r} + \alpha\vec{r'}) = M_{\delta_B}\vec{r} + \alpha M_{\delta_B}\vec{r'} = \vec{c} + \alpha 0 = \vec{c} \ .$$

The secret corresponding with the random string $\vec{r} + \alpha\vec{r'}$ is

$$\vec{1} \cdot (\vec{r} + \alpha\vec{r'}) = s_0 + \alpha\vec{1} \cdot \vec{r'} = s_1 \ .$$

Therefore, the mapping ϕ has the desired properties and the security with respect to unauthorized sets follows. □

We now quote the proof of the other direction. This claim was proved by Beimel and Chor [6, 8] and by van Dijk [52].

Claim 4.9: [6, 8, 52] *Assume there exists a linear scheme over a finite field $\mathcal{K}$ realizing the access structure $\mathcal{A}$ in which the total size of the pieces is d. Then there is a monotone span program over $\mathcal{K}$ computing the function $f_{\mathcal{A}}$ whose size is d.*

Proof: With out loss of generality, assume that $\{0,1\} \subseteq S$. We construct the following span program; for every coordinate of the piece we shall have a row in the span program. This row is labeled by x_i where P_i is the party that the coordinate belongs to. For every pair $\langle s,r \rangle \in \{0,1\} \times R$ we shall have a column in the program. The definition of the span program is the follows: $M_{\langle i,j \rangle, \langle s,r \rangle} = \Pi_{i,j}(s,r)$. That is, the column labeled by $\langle s,r \rangle$ is the vector of pieces generated when the dealer holds the secret s and the random input r. The first $|R|$ columns are indexed by the secret 0 (and the different random strings), and the last $|R|$ columns are indexed by the secret 1. This "vector of secrets" is the objective vector of the program:

$$\vec{v} = \langle \underbrace{0,\ldots,0}_{|R|}, \underbrace{1,\ldots,1}_{|R|} \rangle \ .$$

We claim that this span program computes $f_{\mathcal{A}}$. That is, we will prove that $G \in \mathcal{A}$ if and only if $\vec{v} \in \mathrm{span}(M_{\delta_G})$.

For the first direction, let $G \in \mathcal{A}$. Recall that there exists a linear combination of coordinates of the pieces of the parties in G which equals to the secret. That is, there exist constants $\{\alpha_{i,j}\}$ such that for every $\langle s,r \rangle \in \{0,1\} \times R$ it holds that

$$s = \sum \alpha_{i,j} \Pi_{i,j}(s,r) = \sum \alpha_{i,j} M_{\langle i,j \rangle, \langle s,r \rangle} \ .$$

Now, let $\vec{M}_{\langle i,j \rangle}$ be the $\langle i,j \rangle$ row of M. Hence, rewriting the previous equations we get $\vec{v} = \sum \alpha_{i,j} \vec{M}_{\langle i,j \rangle}$. In other words, $\vec{v} \in \mathrm{span}(M_{\delta_G})$ as required.

For the second direction, assume that there exists a combination of the vectors labeled by variables from G which equals to the objective vector $\vec{v}$. Therefore, applying this combination to their pieces, the parties in G can distinguish when the secret is zero and when the secret is one. Thus, the security requirement implies that $G \in \mathcal{A}$. □

Chapter 5

Lower bounds for Monotone Span Programs

In this chapter we present a new technique for proving lower bounds for monotone span programs, and prove a lower bound of $\Omega(n^{2.5})$ for the 6-clique function. By the results of the previous chapter, this implies the same lower bounds for the total size of the pieces in every linear secret sharing scheme realizing the corresponding access structure.

The chapter is organized as follows. In Section 5.1 we describe our notations, give an application of Nečiporuk's method [87] for lower bounds on the size of (non-monotone) span programs, and a construction of a linear size monotone span program for accepting non-bipartite graphs. In Section 5.2 we present the method for proving the lower bound, and in Sections 5.3 and 5.4 we present applications of the method.

5.1 Preliminaries

A *minterm* of a monotone function is a minimal set of its variables with the property that the value of the function is 1 on any input that assigns 1 to each variable in the set, no matter what the values of the other variables. In this chapter we denote variables by lower case letters, and minterms (sets of variables) by upper case letters, such as A. Script letters, such as $\mathcal{M}$, will be used for families (sets) of sets. We denote by h the number of vertices in a graph, and $\mathcal{K}$ to be a field.

By our definition the size of a span program is the number of rows in the matrix. The number of columns does not effect the size of the span program. However, we observe that it is always possible to use no more columns than the size of the program (since we may restrict the matrix to a set of linearly independent columns without changing the function that is computed).

Observation 5.1: *For every span program $\hat{M}$ there is a span program $\hat{M}'$ which computes the same function has the same number of rows and the number of colums of $\hat{M}'$ is at most*

the number of the rows of $\hat{M}'$.

Following [68] and with Observation 5.1, we can apply Nečiporuk's method [87] to span programs, and get a lower bound of $\Omega(n^{3/2}/\log n)$ for an explicit function with n variables. This is the best lower bound known for the non-monotone span program complexity of an explicit function. Let ED_n be the "element distinctness" function which receives h numbers in the range $\{1, \cdots, h^2\}$ and decides whether all the numbers are distinct. The function ED_n has $n = 2h \log h$ Boolean variables.

Theorem 5.2: $\mathrm{SP}_{\mathrm{GF}(2)}(ED_n) = \Omega(n^{3/2}/\log n)$, *where* $n = 2h \log h$.

Furthermore, Observation 5.1 enables us to count the number of span programs. There are at most $s^{2n}q^{s^2}$ span programs of size s over GF(q). Therefore,

Theorem 5.3: *There exists a Boolean functions* f *such that* $\mathrm{SP}_{\mathrm{GF}(2)}(f) = \Omega(2^{0.5n})$ *and there exists a monotone Boolean functions* f *such that* $\mathrm{mSP}_{\mathrm{GF}(2)}(f) = \Omega(2^{0.5n-o(n)})$.

Using Claim 4.9 we conclude that there exists an access structure for which every linear secret sharing scheme over any field distributes shares of size $2^{\Omega(n)}$.

Next we present a monotone span program of linear size (exactly n) for a function on n variables that is known to have $\Omega(n^{3/2}/(\log n)^3)$ monotone circuit complexity [95, 3, 67]. We consider the function *Non-Bipartite*$_n$, whose input is an undirected graph on h vertices, represented by $n = \binom{h}{2}$ variables, one for each possible edge. The value of the function is 1 if and only if the graph is not bipartite.

Theorem 5.4: $\mathrm{mSP}_{\mathrm{GF}(2)}(\textit{Non-Bipartite}_n) = n$, *where* $n = \binom{h}{2}$.

Proof: We construct a monotone span program over GF(2) accepting exactly the non-bipartite graphs as follows. There will be n rows, each labeled by a variable (possible edge). There is a column for each possible complete bipartite graph on h vertices. The column for a given complete bipartite graph contains the value 0 in each row that corresponds to an edge of the given graph and contains 1 in every other row.

This program rejects every bipartite graph G. This is because G is contained in some complete bipartite graph, and so there will be a column that contains only 0's in the rows labeled by the edges of G. Therefore the vector $\vec{1}$ is not a linear combination of these rows.

Next we show that the program accepts every non-bipartite graph. Since the span program is monotone, it is sufficient to show that it accepts every *minimal* non-bipartite graph, i.e., every odd cycle. Let C be an arbitrary odd cycle. The intersection of any cycle with any complete bipartite graph has an even number of edges. So the odd cycle C has an odd number of edges which are *not* in any given complete bipartite graph. Hence the sum of the row vectors corresponding to all the edges in C is odd in each column, i.e., gives the vector $\vec{1}$ over GF(2), and so C is accepted by the span program. □

We note that the lower bound by Razborov's method (see [95, 3, 67]) for triangles also applies to the function that accepts exactly the non-bipartite graphs, thus the monotone circuit complexity of the function *Non-Bipartite*$_n$ is $\Omega((h/\log h)^3) = \Omega(n^{3/2}/(\log n)^3)$. The gap between monotone span program complexity and monotone circuit complexity was improved by Babai et. al. [4]. They exhibit a function that is computable by monotone span programs whose size is linear but requires monotone circuits of size $n^{\Omega(\log n)}$ and exponential size monotone formulae.

5.2 The Method for Proving Lower Bounds

In this section we present a method for proving lower bounds for monotone span programs. The idea of our method is to show that if the size of a span program (i.e., the number of rows in the matrix) is too small, and the program accepts all the minterms of the function f then it must also accept an input that does not contain a minterm of f, which means that the program does not compute f. Our approach may be viewed as an application of the "fusion method" [67, 96, 111].

We introduce the definition of a critical family of minterms of a monotone Boolean function. We prove that the cardinality of a critical family for a function f is a lower bound on the size of monotone span programs computing f.

Definition 5.5 [critical family]: *Let f be a monotone Boolean function and $\mathcal{M}_f$ be the family of all of its minterms. Let $\mathcal{H} \subseteq \mathcal{M}_f$ be a subfamily of the minterms of f. We say that a subfamily $\mathcal{H} \subseteq \mathcal{M}_f$ is a* critical family for f, *if every $H \in \mathcal{H}$ contains a set $T_H \subseteq H$ such that $|T_H| \geq 2$ and the following two conditions are satisfied.*

C1. The set T_H uniquely determines H in the family $\mathcal{H}$. That is, no other set in the family $\mathcal{H}$ contains T_H.

C2. For any subset $Y \subseteq T_H$, the set $S_Y = \bigcup_{A \in \mathcal{H}, A \cap Y \neq \emptyset} A \setminus Y$ does not contain any member of $\mathcal{M}_f$.

Note that Condition C2 requires that S_Y contains no minterm from f, not just none from $\mathcal{H}$.

Theorem 5.6: *Let f be a monotone Boolean function, and let $\mathcal{H}$ be a critical subfamily of minterms for f. Then for every field $\mathcal{K}$,*

$$\mathrm{mSP}_{\mathcal{K}}(f) \geq |\mathcal{H}| \ .$$

Proof: Let M be the matrix of a monotone span program computing f, and let d be the number of rows of M. Any minterm of $\mathcal{H}$ is accepted by the program. By definition, this means that, for every $H \in \mathcal{H}$, there is some vector $\vec{c}_H \in \mathcal{K}^d$ such that $\vec{c}_H \cdot M = \vec{1}$, and where

$\vec{c}_H$ has nonzero coordinates only at rows labeled by variables from H. (For a given H there may be several such vectors, we pick one of them and denote it by $\vec{c}_H$.)

Since $\vec{c}_H$ is taken from $\mathcal{K}^d$, the number of linearly independent vectors among the vectors $\vec{c}_H$ for $H \in \mathcal{H}$ is a lower bound for d, i.e., for the size of the monotone span program computing f. We show that all the vectors $\vec{c}_H$ for $H \in \mathcal{H}$ must be linearly independent.

Suppose, that this is not the case, i.e., for some $H \in \mathcal{H}$

$$\vec{c}_H = \sum_{A \in \mathcal{A}} \alpha_A \vec{c}_A \ , \tag{5.1}$$

where $\alpha_A \in \mathcal{K}$ and $\mathcal{A} = \mathcal{H} \setminus \{H\}$.

Let us consider the set $T_H \subseteq H$ from Definition 5.5.

Lemma 5.7: *If Equation (5.1) holds then for any nonempty subset $Y \subseteq T_H$ the following must hold.*

$$\sum_{A \in \mathcal{A}, A \cap Y \neq \emptyset} \alpha_A \ = 1 \ .$$

Proof: Suppose that for some $Y \subseteq T_H$, $\sum_{A \in \mathcal{A}, A \cap Y \neq \emptyset} \alpha_A \ = \gamma \neq 1$.

Let us consider the vector

$$\vec{c} = \sum_{A \in \mathcal{A}, A \cap Y \neq \emptyset} \alpha_A \vec{c}_A \ - \vec{c}_H \ . \tag{5.2}$$

We have $\vec{c} \cdot M = (\gamma - 1)\vec{1}$, thus $1/(\gamma - 1)\vec{c} \cdot M = \vec{1}$, and the program accepts the set of variables that label the rows corresponding to nonzero coordinates of $\vec{c}$.

Recall that each $\vec{c}_A$ has nonzero coordinates only at rows labeled by variables from A. Thus for $A \cap Y = \emptyset$ the coordinates of $\vec{c}_A$ are zero at rows labeled by variables from Y. By Equation (5.1),

$$\vec{c} = \vec{0} - \sum_{A \in \mathcal{A}, A \cap Y = \emptyset} \alpha_A \vec{c}_A \ .$$

Therefore, the vector $\vec{c}$ has zero coordinates at all rows labeled by variables from Y.

On the other hand, by Equation (5.2) all the nonzero coordinates of $\vec{c}$ are at rows labeled by variables that appear in some sets A such that $A \cap Y \neq \emptyset$. Therefore, the program accepts $S_Y = \bigcup_{A \in \mathcal{H}, A \cap Y \neq \emptyset} A \setminus Y$, that (by Definition 5.5) does not contain any minterm of f. This proves the lemma. □

From Lemma 5.7, we get a system of linear equations in the unknowns α_A. We prove that this system of equations has no solution, contradicting (5.1). Suppose that $|T_H| = e$. Let us consider the following $(2^e - 1) \times (2^e - 1)$ zero-one matrix N. The rows and columns of N are indexed by the nonempty subsets of T_H, and $N(Y, Z) = 1$ if and only if $Y \cap Z \neq \emptyset$. (This matrix is the complement of the disjointness matrix).

Observation 5.8: *The matrix N has full rank over any field $\mathcal{K}$.*

(This can be shown by a simple transformation of N to a triangular matrix, or by simple induction.)

We will show, that if Equation (5.1) holds then taking $\beta_Z = \sum_{A \in \mathcal{A}, A \cap T_H = Z} \alpha_A$ as a coefficient for the column $Z \neq T_H$, we get the column indexed by T_H as a linear combination of the other columns of N. Notice that the column of N indexed by T_H consists of all 1's. We show that for any Y, $\emptyset \neq Y \subseteq T_H$, we have $\sum_{\emptyset \neq Z \subset T_H} \beta_Z N(Y,Z) = 1$.

By Condition C1 of Definition 5.5, for $A \in \mathcal{A}$ we have $A \cap T_H \neq T_H$. If $Y \subseteq T_H$ then $A \cap Y = A \cap T_H \cap Y$. By Lemma 5.7, if (5.1) holds then we have

$$1 = \sum_{A \in \mathcal{A}, A \cap Y \neq \emptyset} \alpha_A = \sum_{\emptyset \neq Z \subset T_H, Z \cap Y \neq \emptyset} \left(\sum_{A \in \mathcal{A}, A \cap T_H = Z} \alpha_A \right) = \sum_{\emptyset \neq Z \subset T_H} \beta_Z N(Y,Z) \, ,$$

and the column T_H is a linear combination of the other columns of N. Since N has full rank this is not possible, and so (5.1) cannot hold, i.e., all the vectors $\vec{c}_H$ for $H \in \mathcal{H}$ are linearly independent. This concludes the proof of the theorem. □

5.3 Lower bounds for clique functions

We consider the function $Clique_{d,h}$, whose input is an undirected graph on h vertices, represented by $n = \binom{h}{2}$ variables, one for each possible edge. The value of the function is 1 if and only if the graph contains a clique of size d.

It is known ([3, 95]) that the monotone circuit complexity of $Clique_{d,h}$ is $2^{\Omega(\sqrt{d})}$ for $d = O((h/\log h)^{2/3})$, and for fixed d it is $\Omega((h/\log h)^d)$. However, the strongest known lower bound for the monotone span program complexity of the $Clique_{d,h}$ function is our $\Omega(h^5) = \Omega(n^{2.5})$ lower bound that holds for $d \geq 6$ such that $n - d = O(n)$. For $d \leq 4$, we obtain lower bounds that are tight, up to a constant factor.

For a given d, we partition the set of h vertices into d classes $C_1, C_2, \ldots, C_d$ of approximately equal size. We say that a d-clique is *multicolored* if each of its d vertices belong to a different class. Thus a multicolored clique will never contain an edge between two vertices in the same class.

Let $\mathcal{M}$ be an arbitrary family of multicolored d-cliques. Let T_Q be some subset of the edges of a clique $Q \in \mathcal{M}$. Denote the vertices of Q by $v_1, \ldots, v_d$, and consider for $Y \subseteq T_Q$ the set $S_Y = \cup_{G \in \mathcal{M}, G \cap Y \neq \emptyset} G \setminus Y$. Suppose S_Y contains a d-clique Z with vertices $z_1, \ldots, z_d$.

First we present two simple but important observations that are helpful in finding critical families for clique functions.

Claim 5.9: *The vertices of Z all belong to different classes, say $z_i \in C_i$, for $i = 1, \ldots, d$.*

Proof: S_Y only contains edges that appear in d-cliques that belong to the family $\mathcal{M}$, and so contains only edges between vertices from different classes. □

We always list the vertices of a multicolored clique in the order of the partition classes.

Claim 5.10: *For each edge $(v_i, v_j) \in Y$ at least one of $z_i \neq v_i$ or $z_j \neq v_j$ must hold.*

Proof: If Z contained both v_i and v_j for $(v_i, v_j) \in Y$ then Z could not be a d-clique contained in S_Y since S_Y does not contain an edge between v_i and v_j. □

We are ready to construct the critical families.

Lemma 5.11: *For any partition of the h vertices into three classes, the family $\mathcal{M}$ of multicolored 3-cliques is critical for $Clique_{3,h}$.*

Proof: Let Q be an arbitrary multicolored 3-clique (triangle), and let T_Q be the set of two of its edges, for example (v_1, v_2) and (v_2, v_3). There is only one triangle containing T_Q, thus Condition C1 is satisfied. To see that Condition C2 holds, let us consider for $Y \subseteq T_Q$ the set $S_Y = \cup_{G \in \mathcal{M}, G \cap Y \neq \emptyset} G \setminus Y$, and suppose that it contains a triangle Z with vertices z_1, z_2, z_3.

If $Y = T_Q$, then $z_2 = v_2$ must hold, since there are no edges in S_Y incident to any other vertex from C_2. By Claim 5.10 we have $z_1 \neq v_1$ and $z_3 \neq v_3$. Therefore, the edge (z_1, z_3) cannot be present in S_Y, since all the edges of S_Y are contributed by triangles that contain at least one of v_1 or v_3.

If $Y \neq T_Q$, then it consists of a single edge, (v_1, v_2) say. Then S_Y does not contain any edge between the classes C_1 and C_2, and so, by Claim 5.9, cannot contain a triangle. □

Lemma 5.12: *Given any partition of the h vertices into four classes, the family of multicolored 4-cliques is critical for $Clique_{4,h}$.*

Proof: Let Q be an arbitrary multicolored 4-clique, and let T_Q be the set of two of its nonadjacent edges, for example (v_1, v_2) and (v_3, v_4). Condition C1 is satisfied, since two nonadjacent edges uniquely determine a 4-clique. To see that Condition C2 holds, as in the previous lemma, let us consider S_Y for $Y \subseteq T_Q$ and suppose that it contains a 4-clique Z with vertices z_1, z_2, z_3, z_4.

If $Y = T_Q$ then, by Claim 5.10, without loss of generality we have $z_1 \neq v_1$. Any edges incident to z_1 could only be contributed to S_Y by cliques that contain (v_3, v_4). Thus, a clique containing z_1 would also have to contain both v_3 and v_4, which is not possible by Claim 5.10.

As in the previous lemma, if $Y \neq T_Q$ then it consists of a single edge and S_Y does not contain a 4-clique. □

We note that for $d \geq 5$ the family of multicolored d-cliques is not critical for $Clique_{d,h}$. For example, for $k = 5$, any choice of T_H for a multicolored 5-clique H with vertices v_1, v_2, v_3, v_4, v_5, must contain either the set $Y_1 = \{(v_1, v_2), (v_1, v_3), (v_4, v_5)\}$ or $Y_2 = \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_1, v_5)\}$, up to renaming of the vertices. Each of the sets S_{Y_1} and S_{Y_2} contain the multicolored 5-clique on vertices v_1, z_2, z_3, z_4, z_5, where $z_i \neq v_i$ for $i = 2, \ldots 5$.

The critical families we use for proving lower bounds for 5- and 6-cliques will be appropriately chosen subfamilies of multicolored cliques.

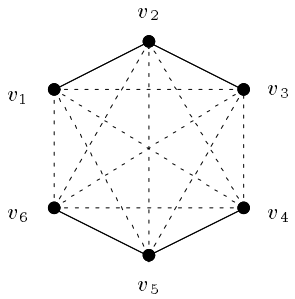

Figure 5.1: Illustrations of the set T_H
T_H איור 5.1: תאור של הקבוצה

Theorem 5.13: *For every field $\mathcal{K}$,*

$$\mathrm{mSP}_{\mathcal{K}}(Clique_{6,h}) = \Omega(n^{2.5}) \ .$$

Proof: We show that the family of minterms of the $Clique_{6,h}$ function contains a large critical subfamily $\mathcal{Q}$. Let us assume that $h = 6q$, and partition the set of h vertices into six classes of size q. $\mathcal{Q}$ will be a subfamily of the multicolored 6-cliques, under this partition.

For members of $\mathcal{Q}$, we restrict the edges allowed between vertices in the classes C_1 and C_3, and similarly between the classes C_4 and C_6. The *legal* pairs of vertices which we allow to be connected by an edge will be specified by a $q \times q$ Boolean matrix D. Between all other pairs of classes we allow arbitrary edges. The edge (a, b) with $a \in C_1$ and $b \in C_3$ ($a \in C_4$ and $b \in C_6$, respectively) is allowed in a member of $\mathcal{Q}$ if and only if $D(a, b) = 1$. We choose D such that it does not contain any complete (all ones) 2×2 submatrices. For example the incidence matrix of a projective plane has this property, and its number of one is $\Theta(q^{3/2})$, with $\Theta(q^{1/2})$ ones in each row and column. The constructions in [70, 88] can also be used. (The construction of matrices with similar properties for arbitrary q is described in [91].)

The family $\mathcal{Q}$ consists of all the multicolored 6-cliques that satisfy the restriction on the edges between classes C_1 and C_3, and between C_4 and C_6. The number of such 6-cliques is $\Theta(q^5)$, thus we have $|\mathcal{Q}| = \Theta(q^5) = \Theta(n^{2.5})$.

Next we show that $\mathcal{Q}$ is critical for $Clique_{6,h}$. Consider any $Q \in \mathcal{Q}$, and denote its vertices by $v_1, \ldots, v_6$. The set T_Q we choose will consist of the four edges (v_1, v_2), (v_2, v_3), (v_4, v_5), (v_5, v_6). The critical family is illustrated in Figure 5.2. Obviously, Condition C1 is satisfied.

We now prove that Condition C2 holds. For $Y \subseteq T_Q$, suppose the set $S_Y = \cup_{G \in \mathcal{Q}, G \cap Y \neq \emptyset} G \setminus Y$ contains a 6-clique Z with vertices $z_1, \ldots, z_6$. We analyze the different cases for Y. Case 1. Let $Y = T_Q$. Notice that if both $z_2 \neq v_2$ and $z_5 \neq v_5$, then S_Y does not contain an edge between z_2 and z_5, thus we have $z_2 = v_2$ or $z_5 = v_5$. These possibilities are illustrated in Figure 5.2.

Suppose that only one of these equalities holds, for example $z_2 = v_2$ but $z_5 \neq v_5$. Then, by Claim 5.10, $z_1 \neq v_1$ and $z_3 \neq v_3$ must hold. The edge (z_1, z_5) can only be contributed to S_Y by a clique that contains the edge (v_2, v_3), and similarly the edge (z_3, z_5) can only be contributed to S_Y by a clique that contains the edge (v_2, v_1). This means that the edges $(z_1, v_3), (v_1, z_3)$ as well as the edges (v_1, v_3) and (z_1, z_3) appear in some member of the family $\mathcal{Q}$. However, this is not possible by our restriction on the legal edges between C_1 and C_3.

Suppose now that both $z_2 = v_2$ and $z_5 = v_5$ holds. Then by Claim 5.10 we have $z_1 \neq v_1$, $z_3 \neq v_3$, $z_4 \neq v_4$ and $z_6 \neq v_6$. The edge (z_1, z_4) can only be contributed by a clique that contains (v_2, v_3) or (v_5, v_6). This means that at least one of the edges (z_4, v_6) or (z_1, v_3) is legal. Similarly, from the presence in Z of the edges (z_1, z_6), (z_3, z_4) and (z_3, z_6), respectively, we know that at least one each of (v_4, z_6) or (z_1, v_3), (z_4, v_6) or (v_1, z_3), and (v_4, z_6) or (v_1, z_3), respectively, are legal edges. This means that either both (z_4, v_6) and (v_4, z_6) or both (z_1, v_3) and (v_1, z_3) are legal, and since (v_i, v_j) and (z_i, z_j) must be legal for all i, j, we get a contradiction with our restriction on the possible edges of members from $\mathcal{Q}$.

Case 2. Let $Y \neq T_Q$. In this case the edges in Y cover t vertices, $2 \leq t \leq 5$. We show that S_Y does not even contain a t-clique on the t classes involved. For $t \leq 4$ this follows directly from Lemma 5.11 and Lemma 5.12.

We still have to deal with the case when $t = 5$, which can only happen if Y consists of three edges. Suppose (without loss of generality) that the three edges of Y are (v_1, v_2), (v_2, v_3) and (v_4, v_5). If $z_2 \neq v_2$, then all the edges incident to z_2 could only be contributed to S_Y by cliques that contain (v_4, v_5). That would mean that the only vertices in C_4 and C_5 connected to z_2 in S_Y are v_4 and v_5. Thus we could not get a 6-clique in S_Y that contains z_2. Therefore, $z_2 = v_2$ must hold. Then we have by Claim 5.10 that $z_1 \neq v_1$, $z_3 \neq v_3$ and, without loss of generality, $z_5 \neq v_5$. We get a contradiction with the restriction on the edges between C_1 and C_3 as in Case 1.

We have proved that Condition C2 is also satisfied, and $\mathcal{Q}$ is a critical family for f. The lower bound follows from Theorem 5.6. □

Theorem 5.14: *For every field $\mathcal{K}$,*

$$\begin{array}{lrcccc} \textit{for every } 6 \leq d \leq h & \mathrm{mSP}_{\mathcal{K}}(\mathit{Clique}_{d,h}) & = & \Omega((h-d)^5) & = & \Omega((n^{0.5}-d)^5)\ , \\ & \mathrm{mSP}_{\mathcal{K}}(\mathit{Clique}_{5,h}) & = & \Omega(h^{4.5}) & = & \Omega(n^{2.25})\ , \\ & \mathrm{mSP}_{\mathcal{K}}(\mathit{Clique}_{4,h}) & = & \Theta(h^4) & = & \Theta(n^2)\ , \\ & \mathrm{mSP}_{\mathcal{K}}(\mathit{Clique}_{3,h}) & = & \Theta(h^3) & = & \Theta(n^{1.5})\ . \end{array}$$

The proof of this theorem is basically included in the proof of the lower bound for 6-cliques and in Lemma 5.11 and Lemma 5.12.

5.4 A function with minterms of size 2

Access structure in which the size of every minimal reconstructing set is 2 have received much attention (e.g. [27, 28, 30, 36, 52, 53, 107]). In [53] it is proved that there exists

such an explicit access structure for which the total size of the pieces in every secret-sharing scheme is $\Omega(n \log n)$ times the length of the secret (for every finite set of possible secrets). That is, the monotone span program complexity of the function that represents the access structure is $\Omega(n \log n)$. In this section we exhibit an explicit function whose minterms are of size 2 and whose monotone span program complexity $\Omega(n^{3/2})$. I.e., the total size of shares in every linear secret sharing scheme for the corresponding access structure is at least $\Omega(n^{3/2})$. Let $L_1, \ldots, L_h$ be h subsets of $\{1, \ldots, h\}$ such that the size of the intersection of every two subsets is at most 1. For example, the lines of a projective plane can be used. Given the sets $L_1, \ldots, L_h$, we define the function *Lines*, which has $n = 2h$ variables denoted $\{a_1, \ldots, a_h, b_1, \ldots, b_h\}$, and whose minterms are $\{\{a_i, b_j\} : j \in L_i\}$.

Theorem 5.15: *For every field $\mathcal{K}$,*

$$\mathrm{mSP}_{\mathcal{K}}(Lines) \geq \sum_{i=1}^{h} |L_i| \ .$$

Proof: We prove that the family of all minterms of the function *Lines* is a critical family for *Lines*. The set T_H for every minterm H is simply H, and so Condition C1 is obviously satisfied.

To prove Condition C2, we take an arbitrary minterm, say $\{a_1, b_1\}$ without loss of generality, and consider the set $X = S_{\{a_1,b_1\}} = \{b_j : j \in L_1\} \cup \{a_i : 1 \in L_i\} \setminus \{a_1, b_1\}$. Suppose that there is some minterm $\{a_i, b_j\}$ contained in X. Now $1 \in L_i$ since $a_i \in X$, and $j \in L_1$ since $b_j \in X$. We also have $1 \in L_1$ since $\{a_1, b_1\}$ is a minterm, and $j \in L_i$ since $\{a_i, b_j\}$ is a minterm. However $j \neq 1$, and this contradicts the fact that the size of the intersection of L_1 and L_i is at most 1. Obviously, the sets $S_{\{a_1\}}$ and $S_{\{b_1\}}$ do not contain any minterms either. □

Using the lines of a projective plane or the constructions from [70, 88, 91] for the sets $L_1, \ldots, L_h$ we have $\sum_{i=1}^{h} |L_i| = h^{3/2} + O(h)$. There exists a monotone formula for this function of size $n^{3/2} + O(n)$ (take the DNF formula with a term for every minterm and group the terms that include each a_i). Thus, we show an asymptotically matching upper bound for this function.

Corollary 5.16: *For some explicitly given sets $L_1, \ldots, L_h$ we have, for every field $\mathcal{K}$,*

$$\mathrm{mSP}_{\mathcal{K}}(Lines) = h^{3/2} + O(h) = \Theta(n^{3/2}) \ .$$

Since every monotone function with minterms of size 2 has a DNF representation of size $O(n^2)$, such a function has a monotone span program of size $O(n^2)$. However, using Theorem 5.6 one can prove only lower bounds of size at most $\Omega(n^{1.5})$ for function with minterms of size 2 (since every critical family of minterms of size two is of size $O(n^{1.5})$). It is an open problem whether one can construct more efficient monotone span programs for function with minterms of size 2. If this is not possible then this would prove that the lower bounds proved using Theorem 5.6 are not tight.

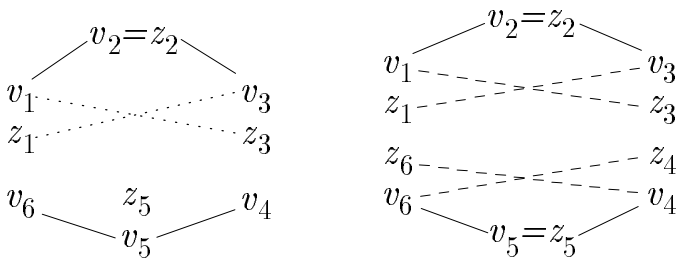

Figure 5.2: Illustrations for Case 1 of Theorem 5.13

איור 5.2: תאור של מקרה 1 של משפט 5.13

Chapter 6

Communication in Key Distribution Schemes

In this chapter, we study the relationships between communication and space efficiency of key distribution schemes. We prove that communication does not help in *unrestricted schemes*. On the other hand, we show that for restricted schemes, which are secure only when used by a limited number of conferences, communication can substantially improve the space efficiency. Furthermore, we prove lower bounds on the space efficiency of restricted schemes.

This chapter is organized as follows. In Section 6.1 we consider some properties of non-communicating key distribution schemes. In Section 6.2 we present our proof of the lower bound on the efficiency for the weak non-communicating schemes. In Section 6.3 we use this result to prove a lower bound for unrestricted communicating schemes. In Section 6.4 we prove lower bounds for restricted communicating schemes, and in Section 6.5 we describe some efficient constructions of restricted schemes.

6.1 Properties of Non-Communicating Schemes

In Definition 3.7 we defined non communicating (g, b) key distribution schemes. We give an equivalent condition for the security of these schemes. The equivalence of the conditions follows directly from Bayes rule.

Lemma 6.1: *A non communicating (g, b) key distribution scheme is secure if and only if for every conference G of size g, every disjoint coalition B of size b, every vector of pieces $\langle u_1, \ldots, u_n \rangle$ (that is dealt with positive probability), and every key $k \in K$*

$$\Pr\left[\, k_G(r) = k \,\middle|\, \bigwedge_{P_j \in B} (\mathcal{U}_j(r) = u_j) \,\right] = \Pr\left[\, k_G(r) = k \,\right] \triangleq \mathcal{P}(k) \ . \tag{6.1}$$

The previous lemma states that the key of a conference G is a random variable which is statistically independent of the pieces of a disjoint coalition of size b. However, it is not guaranteed that the keys of different conferences are statistically independent of each other given the pieces of a disjoint coalition of size b. In Claim 6.2 we give a condition in which the independence of some keys are guaranteed. This claim is used in Section 6.5 for constructing one-time key distribution schemes and in Section 7.3 for constructing one-time secret sharing schemes with public reconstruction. We show that if we start with a non-communicating $(2, g+b-2)$ key distribution scheme then that the collection of keys of all pairs of parties in a set G of g parties is uniformly distributed and independent of the pieces of coalition B of b parties.

Claim 6.2: *Let $\mathcal{U}$ be a non-communicating $(2, g+b-2)$-scheme with a-priori distribution $\mathcal{P}$ on a key domain K. Let G and B be sets of g and b parties respectively, such that $G \cap B = \emptyset$, $G_1, G_2, \ldots, G_{\binom{g}{2}}$ be all the subsets of G of cardinality two, $k_1, \ldots, k_{\binom{g}{2}}$ be any combination of $\binom{g}{2}$ keys from K, and $\langle u_1, \ldots, u_n \rangle$ be a vector of pieces that is distributed with positive probability. Then:*

$$\Pr[\bigwedge_{1 \le i \le \binom{g}{2}} k_{G_i}(r) = k_i \mid \bigwedge_{P_j \in B} \mathcal{U}_j(r) = u_j] = \prod_{1 \le i \le \binom{g}{2}} \mathcal{P}(k_i) \ . \tag{6.2}$$

Proof: To simplify the notations, we denote $t = \binom{g}{2}$, and denote the key of the set G_i, i.e. $k_{G_i}(r)$, by $k_i(r)$. Notice that

$$\Pr[\bigwedge_{1 \le i \le t} k_i(r) = k_i \mid \bigwedge_{P_j \in B} \mathcal{U}_j(r) = u_j] = \prod_{1 \le h \le t} \Pr[k_h(r) = k_h \mid \bigwedge_{h+1 \le i \le t} k_i(r) = k_i \wedge \bigwedge_{P_j \in B} \mathcal{U}_j(r) = u_j]$$

Therefore, to prove Equation (6.2) (and therefore the claim) it is enough to prove that for every h, where $1 \le h \le t$, it holds that:

$$\Pr[\, k_h(r) = k_h \mid \bigwedge_{h+1 \le i \le t} k_i(r) = k_i \wedge \bigwedge_{P_j \in B} \mathcal{U}_j(r) = u_j \,] = \mathcal{P}(k_h) \ . \tag{6.3}$$

That is, the key $k_h(r)$ is independent of the other keys and the pieces of the parties in B. To prove Equation (6.3), we partition the conditional probability space into disjoint subspaces, and prove this equality for every subspace. Thus, Equation (6.3) will hold for the original conditional space. The partition of the conditional space is according to the pieces of the $g-2$ parties in $G \setminus G_h$. These pieces determine the keys of all the G_i except G_h. Without loss of generality $G_h = \{P_1, P_2\}$. Let $\vec{v} = \langle v_3, \ldots, v_{g+b} \rangle$, be any vector of pieces that is distributed with positive probability, such that:

- For every $P_j \in B$ it holds that $u_j = v_j$.
- For every i, such that $h+1 \le i \le t$, it holds that the key of G_i reconstructed according to the pieces of $\vec{v}$ is k_i.

Since the cardinality of $B \cup G \setminus G_h$ is $g+b-2$ then the security property of the $(2, g+b-2)$ scheme guarantees that

$$\Pr[\, k_h(r) = k_h \mid \bigwedge_{P_j \in G \setminus G_h} \mathcal{U}_j(r) = v_j \wedge \bigwedge_{P_j \in B} \mathcal{U}_j(r) = v_j \,] = \mathcal{P}(k_h)\,. \qquad \square$$

We now consider a weakening of the security requirement of key distribution schemes. Instead of requiring that the conditional probability (given any pieces of a bad set B) of every key equals the a-priory probability, we will only require that this conditional probability is positive. To simplify this discussion, we will only consider non-communicating weak schemes.

Definition 6.3 [Weak Key Distribution Schemes]: *A* weak non-communicating (g, b) key distribution scheme *is a non-communicating (g, b) scheme in which the security property is relaxed:*

Weak security property *Let B be a coalition of b (bad) parties, and let G be a conference of g (good) parties, such that $G \cap B = \emptyset$. Then the parties in B, having their pieces, cannot rule out any key of G. In other words, for every vector of pieces $\vec{u} = \langle u_1, \ldots, u_n \rangle$ which is dealt with positive probability, and every possible key $k \in K$, there exists a vector of pieces $\vec{u'}$ that agrees with $\vec{u}$ on the pieces of B, but the key of the set G according to the vector $\vec{u'}$ is k. Formally,*

$$\Pr\,[\, k_G(r) = k \mid \bigwedge_{P_j \in B} \mathcal{U}_j(r) = u_j \,] > 0$$

where the probability is taken over the random input of the dealer.

By Lemma 6.1, it follows that unrestricted non-communicating schemes are a special case of weak non-communicating schemes. Therefore, every lower bound for weak schemes implies the same lower bound for unrestricted non-communicating schemes. We claim that the weak security requirement is *not* reasonable, since every "bad" coalition B could gain a lot of information. The reason we do define weak schemes is because we show that the lower bounds on the size of the pieces hold even for these weak schemes.

6.2 Lower Bounds for Non-Communicating Schemes

Blundo et. al. [29] prove a tight lower bound on the size of the pieces in every non-communicating key distribution scheme. Their proof is based on the entropy function, and, in our opinion, does not reveal the intuition behind this lower bound. We present a simpler proof of this lower bound, which is not based on entropy. Furthermore, in our proof we only use the weak security requirement. That is, even weak (g, b) schemes, where the entropy of the keys may be very low, must have large domain of pieces. Thus, our proof yields a stronger result than the lower bound of [29]. We use this stronger result in the sequel.

Theorem 6.4 [29]: *Let $\mathcal{U}$ be a weak non-communicating (g,b) scheme with n parties and domain of keys K. Let U_i be the domain of pieces of party P_i in $\mathcal{U}$. Then for every i $(1 \leq i \leq n)$:*

$$|U_i| \geq |K|^{\binom{g+b-1}{g-1}} .$$

That is, the size of the pieces is at least $\binom{g+b-1}{g-1}$ times the size of the key.

Proof: Consider a (g,b) scheme with a domain of keys K. Without loss of generality, we assume that there are *exactly* $g+b$ parties, which we denote by $\{P_1,\ldots,P_{g+b}\}$. We prove the lower bound on the domain of pieces of party P_1. Let $G_1,\ldots,G_\ell$ be all the sets of cardinality g that contain party P_1, where $\ell \triangleq \binom{g+b-1}{g-1}$. Let $\vec{k} = \langle k_1, k_2, \ldots, k_\ell\rangle$ be any vector in K^ℓ. We claim that there exists a vector of pieces $\vec{u} = \langle u_1,\ldots,u_{g+b}\rangle$ (which is dealt with positive probability), such that for every $1 \leq i \leq \ell$ the key of the set G_i reconstructed from the pieces in $\vec{u}$ equals k_i. Otherwise, let i $(i \leq \ell)$ be a maximal index such that there exist keys $k'_i,\ldots,k'_\ell \in K$ (where $k'_i \neq k_i$) and there exists a vector of pieces $\vec{u'} = \langle u'_1,\ldots,u'_{g+b}\rangle$, which is dealt with positive probability, such that the keys in $\vec{k'} \triangleq \langle k_1,\ldots,k_{i-1},k'_i,\ldots,k'_\ell\rangle$ are reconstructed from the pieces in $\vec{u'}$. Such index $i \geq 2$ exists since for $i=1$, there is some vector of pieces from which G_1 reconstructs the key k_1. Consider the set $B = \{P_1,\ldots,P_{g+b}\}\setminus G_i$, which contains exactly b parties. Since the set B intersects G_j for every P_j such that $j \neq i$, the parties in B can reconstruct the keys of the conferences $G_1,\ldots,G_{i-1},G_{i+1},\ldots,G_\ell$. Therefore, the pieces from $\vec{u'}$ of the parties in B determine that the reconstructed keys of the conferences $G_1,\ldots,G_{i-1}$ are $k_1,\ldots,k_{i-1}$, respectively. Suppose there is a positive probability that G_i's key equals k_i, given the pieces u'_i. Then in particular there is a vector of pieces giving rise with non-zero probability to the keys $k_1,k_2,\ldots,k_{i-1},k_i$. This contradicts the choice of i as the maximal index. Therefore,

$$\Pr[\, k_{G_i}(r) = k_i \mid \bigwedge_{P_j \in B} \mathcal{U}_j(r) = u'_j \,] = 0$$

But this violates the weak security property of the (g,b) scheme, a contradiction to our assumption.

Hence for every $\vec{k} \in K^\ell$, there is a vector of pieces $\vec{u}$ for the parties, in which the vector of reconstructed keys for the sets $G_1,\ldots,G_\ell$ is $\vec{k}$. Since party P_1 computes the keys of the sets $G_1,\ldots,G_\ell$ from his piece, it follows that his piece must be different for every pair of different vectors of keys for the sets $G_1,\ldots,G_\ell$. There are $|K|^\ell$ possible vectors of keys, therefore there are at least $|K|^\ell$ different pieces for party P_1. That is, $|U_1| \geq |K|^\ell = |K|^{\binom{g+b-1}{g-1}}$, as claimed. □

We remark that if the keys of all sets were independent random variables then using the same ideas of this proof, we can prove a lower bound of $|K|^{\binom{n}{g-1}}$. Another observation is that we can consider a key distribution scheme in which only some pre-defined subsets of size g can reconstruct a key. Our proof actually supplies a lower bound for this setting as well.

Lemma 6.5: *Let $\mathcal{U}$ be a (weak) non-communicating (g,b) scheme with exactly $g+b$ parties and domain of keys K, in which party P_i is a member of at least ℓ sets that can reconstruct a key. Let U_i be the domain of pieces of P_i in $\mathcal{U}$ then*

$$|U_i| \geq |K|^{\ell} .$$

Notice that ℓ can be at most $\binom{g+b-1}{g-1}$.

Using symmetric degree b multinomials with g variables, Blundo et. al [29] have constructed an unrestricted non-communicating (g,b) scheme with domains of pieces $|U_i| = |K|^{\binom{g+b-1}{g-1}}$ with uniform a-priori distribution on the keys (provided that $|K| \geq n$ and $|K|$ is a prime power). So, the lower bound is tight (except for small domains of keys).

6.3 Removing the Communication from Unrestricted Schemes

In this section we show how to transform an unrestricted communicating scheme into a weak non-communicating key distribution scheme, without enlarging the domain of pieces. Therefore, the lower bound on the cardinality of the domain of pieces applies to unrestricted communicating schemes as well.

Theorem 6.6: *Let $\mathcal{U}$ be an unrestricted communicating (g,b) key distribution scheme with $n \geq g+b$ parties and domain of keys K. Let $U_1, \ldots, U_n$ be the domains of pieces of the parties in $\mathcal{U}$. Then for every party P_i:*

$$|U_i| \geq |K|^{\binom{g+b-1}{g-1}} .$$

That is, the size of the pieces is at least $\binom{g+b-1}{g-1}$ times the size of the key.

Proof: The idea of the proof is to fix, for every set G of g parties, a possible communication C_G (i.e. one that is exchanged with positive probability when G communicates in order to generate a conference key). The vector of all these communications is public knowledge. Now the dealer deals only vectors of pieces that are consistent with all the communications C_G's. When a member of a set G wishes to determine a conference key, he applies the reconstruction function to his piece and the fixed communication C_G. This way, no communication is required. In the rest of proof, we show first how to choose communications for different conferences such that they are consistent among themselves. This implies the existence of vectors of pieces that are consistent with all the communications. Once this is done, it is clear that the non-communicating scheme has the reconstruction property. We then prove that the resulting non-communicating scheme has the weak security property. Therefore, it is

a weak non-communicating (g, b) scheme.[1] By Theorem 6.4 the cardinality of the domain of pieces of every party in the resulting weak non-communicating scheme is at least $|K|^{\binom{g+b-1}{g-1}}$. But the domain of the pieces in the non-communicating scheme is not larger than that of the communicating scheme. Therefore, the lower bound on the size of the pieces applies to the original communicating scheme as well.

To complete the proof, we first show how to choose a set of communications C_G (for all G's) in a consistent way. To do this, we first fix an arbitrary vector of pieces $\vec{u}$ which is dealt by the dealer in the original scheme with positive probability. We also fix the local random input of each party. Each communication C_G is the one determined when the parties of G hold pieces from $\vec{u}$, and have the fixed random inputs. It is clear that $\vec{u}$ is consistent with all these conversations. The dealer in the new scheme chooses at random a vector of pieces that is consistent with the communications. That is, the dealer chooses a vector of pieces from the (non-empty) set of vectors of pieces $\vec{v}$ for which there exists a vector of local random inputs $\vec{r}$ for the parties, such that every conference G of g parties, holding the pieces of $\vec{v}$, and having the random inputs $\vec{r}_G$, communicate C_G (where $\vec{r}_G$ is the restriction of $\vec{r}$ to the members of G).

We now show that the resulting non-communicating scheme is weakly secure. Let G be any conference of cardinality g, and B be a disjoint coalition of cardinality b. By the security property of the communicating scheme, it follows that for every vector of pieces that is consistent with the fixed conversations, and every key $k \in K$, there exists a vector of pieces which is consistent with the given pieces of the parties in B, such that the reconstructed key of conference G equals k. That is, the non-communicating scheme has the weak security property, as claimed. □

We can define the notion of weak security for unrestricted communicating schemes in a similar manner to Definition 6.3. The lower bound of Theorem 6.6 is also applicable to such weak unrestricted communicating schemes.

6.4 Lower Bounds for Restricted Schemes

By Theorem 6.6, communication cannot decrease the size of the pieces of information given to the parties in unrestricted key distribution schemes. In order to decrease the size of the pieces of information, we consider restricted schemes in which the key distribution schemes should be secure only for a restricted number of conferences. Which conference will generate a key is not known a-priori, so the distributed pieces should accommodate any combination of conferences (up to the limit on their number).

[1] In this proof we do not define the probability distribution under which the dealer distributes the consistent vectors of pieces. We only require that every consistent vector is distributed with positive probability. It is possible to define a probability distribution on the consistent vectors, such that the induced (g, b) scheme will have the unrestricted security property.

The proof that unrestricted communicating schemes and unrestricted non-communicating schemes have the same space efficiency (Theorem 6.6) is not applicable for restricted schemes. It is possible, for example, to transform a one-time secure communicating scheme into a non-communicating scheme, using the technique of Theorem 6.6. However, this would yield a non-communicating scheme which is secure only with respect to a single *fixed* conference, depending on the one initiating the communication. Indeed, in Section 6.5 we show that restricted schemes can be much more efficient than unrestricted schemes. In this section we give lower bounds on their efficiency.

We first give a simple lower bound for every τ-restricted (g, b) key distribution scheme. We show that for $\tau \leq \binom{g+b-1}{g-1}$ the cardinality of the domains of pieces in any τ-restricted (g, b) key distribution scheme is at least $|K|^\tau$. Therefore, for $\tau \geq \binom{g+b-1}{g-1}$ the unrestricted non-communicating scheme of [29] is space optimal even with respect to τ-restricted schemes. In addition, this shows that for restricted schemes with smaller τ, some dependence on the size of τ is unavoidable. We improve this bound for the case $\tau \leq (b/g)^g$. In particular, for one time schemes we prove a $|K|^{1+\lfloor b/(g-1)\rfloor}$ lower bound (the upper bound is $|K|^{2+2(b-1)/g}$).

We first present a simple lower bounds for τ-restricted schemes.

Lemma 6.7: *Let $\tau \leq \binom{g+b-1}{g-1}$. In every τ-restricted (g, b)-scheme with $n \geq g + b$ parties and domain of keys K, the cardinality of the domain of pieces of every party is at least $|K|^\tau$.*

Proof: Again, we limit the number of parties to $g+b$. Using the same ideas as in the proof of Theorem 6.6, we transform a τ-restricted (g, b)-scheme into a weak non-communicating (g, b)-scheme in which τ pre-defined sets can reconstruct a key. That is, we fix consistent conversations of the τ sets. The dealer generates vectors of pieces consistent with these conversations. The original scheme is secure for τ conferences, therefore by fixing τ conversations, we get a secure scheme in which these τ sets can reconstruct a key without any communication. Since $\tau \leq \binom{g+b-1}{g-1}$, there are τ sets that contain P_i. Choosing τ sets containing P_i as the pre-defined sets, we apply Lemma 6.5 to the transformed scheme. By this lemma the cardinality of the domain of pieces of P_i in the transformed scheme at least $|K|^\tau$. By the transformation, the cardinality of the domain of pieces in the transformed scheme is at most the cardinality of the domain of pieces in the τ-restricted scheme. Therefore, the cardinality of the domain of pieces of every party in the τ-restricted scheme is at least $|K|^\tau$. □

We now improve the previous lower bound for schemes in which $\tau \leq (b/g)^g$. The proof of this lower bound uses entropy and mutual information. For definitions of these information theoretic terms, the reader can refer to [42, 45, 60] (see also Appendix A). In the proof we use the following claim of Maurer [77] and Ahlswede and Csiszar [2]. Its context is a system where two (coin flipping) parties, each with private piece of information, execute a protocol by communicating over a public channel. After the execution of the protocol, the two parties generate a common key, such that a third party overhearing all the communication does not have any information on the key. First, the claim states that the conditional mutual

information of the pieces (measuring the information known to the two parties but not to the third party) held by the two parties cannot be increased after a conversation on a public channel. Second, the claim states that the conditional mutual information of the initial pieces is at least the entropy (uncertainty) of the generated key. In the claim $\mathcal{U}_i$ is the random variable which denotes the piece of P_i, and K is a random variable which denotes the key.

Claim 6.8 [2, 77]: *Let $\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3$ be random variables, held by parties P_1, P_2, P_3 respectively. Parties P_1 and P_2 communicate on a public channel according to some protocol. Denote the communication by the random variable M. Then, $I(\mathcal{U}_1 ; \mathcal{U}_2 \mid \mathcal{U}_3) \geq I(\mathcal{U}_1 ; \mathcal{U}_2 \mid M\mathcal{U}_3)$. Furthermore, assume that after the execution of the protocol parties P_1 and P_2 agree on a key K known to both of them, such that P_3 has no information on the key. That is, $H(K \mid \mathcal{U}_1, M) = H(K \mid \mathcal{U}_2, M) = 0$, while $H(K \mid \mathcal{U}_3, M) = H(K)$. Then, $I(\mathcal{U}_1 ; \mathcal{U}_2 \mid \mathcal{U}_3) \geq H(K)$.*

We first consider $(2, b)$-schemes. Recall that in the non-communicating $(2, b)$-scheme of Blom [24], the cardinality of the domain of pieces is $|K|^{b+1}$. In Lemma 6.9 we prove that $|K|^{b+1}$ is a lower bound even for *one-time* $(2, b)$-schemes. Thus, in this case communication does not help even if we consider only one-time schemes.

Lemma 6.9: *Let $\mathcal{U}$ be a one time $(2, b)$ scheme with $n \geq b + 2$ parties and domain of keys K. Assume that the a-priory probability of each key in K is uniform. Then the cardinality of the domain of pieces of every party in the scheme $\mathcal{U}$ is at least $|K|^{b+1}$.*

Proof: Without loss of generality, we prove that $|U_1| \geq |K|^{b+1}$, where U_1 is the domain of pieces of P_1 in the scheme $\mathcal{U}$. We start with a one-time $(2, b)$ scheme $\mathcal{U}$. For every $i > 2$, we construct a new scheme with 3 parties. In the new scheme P_1 receives $\mathcal{U}_1$, the piece of P_1 in $\mathcal{U}$, party P_2 receives $\mathcal{U}_i$ the piece of P_i in $\mathcal{U}$, and party P_3 receives the pieces from $\mathcal{U}$ of the parties $P_{i+1}, \ldots, P_{b+2}$. In the new scheme parties P_1 and P_2 can generate the key of the old parties $\{P_1, P_i\}$. Since party P_3 receives at most b pieces of the old scheme, he has no information on the generated key. Therefore, we are in the scenario of Claim 6.8. We conclude that for every i $(2 \leq i \leq b + 2)$:

$$\mathrm{I}(\mathcal{U}_1 ; \mathcal{U}_i \mid \mathcal{U}_{i+1} \ldots \mathcal{U}_{b+2}) \geq \mathrm{H}(K) .$$

On the other hand, by definition

$$\begin{aligned} \sum_{i=2}^{b+2} \mathrm{I}(\mathcal{U}_1 ; \mathcal{U}_i \mid \mathcal{U}_{i+1} \ldots \mathcal{U}_{b+2}) &= \sum_{i=2}^{b+2} \mathrm{H}(\mathcal{U}_1 \mid \mathcal{U}_{i+1} \ldots \mathcal{U}_{b+2}) - \sum_{i=2}^{b+2} \mathrm{H}(\mathcal{U}_1 \mid \mathcal{U}_i \mathcal{U}_{i+1} \ldots \mathcal{U}_{b+2}) \\ &= \mathrm{H}(\mathcal{U}_1) - \mathrm{H}(\mathcal{U}_1 \mid \mathcal{U}_2 \ldots \mathcal{U}_{b+2}) \\ &\leq \mathrm{H}(\mathcal{U}_1) . \end{aligned}$$

Combing the last two arguments together, we deduce that

$$\mathrm{H}(\mathcal{U}_1) \geq \sum_{i=2}^{b+2} \mathrm{I}(\ \mathcal{U}_1\ ;\ \mathcal{U}_i\ \mid \mathcal{U}_{i+1} \ldots \mathcal{U}_{b+2}\ ) \geq (b+1)\mathrm{H}(K)\ .$$

It holds that $\mathrm{H}(\mathcal{U}_1) \leq \log |U_1|$, and since we assume uniform distribution on K then $\mathrm{H}(K) = \log |K|$. Thus,

$$\log |U_1| \geq \mathrm{H}(\mathcal{U}_1) \geq (b+1)\mathrm{H}(K) = (b+1)\log |K|\ ,$$

which yields the claimed bound $|U_1| \geq |K|^{b+1}$. □

The proof of Lemma 6.9 (with respect to the domain U_1) does not require that every pair of parties should be able to reconstruct a key. It only requires that P_1 and every other party can generate a key. It is also interesting to notice that any scheme meeting this lower bound satisfies $\mathrm{H}(\ \mathcal{U}_1\ \mid \mathcal{U}_2 \ldots \mathcal{U}_{b+2}\ ) = 0$, so every $b+1$ parties can reconstruct the pieces of all the other parties. We now use Lemma 6.9 to prove a lower bound for every conference size g and various values of τ.

Theorem 6.10: *Let τ, g, b be positive integers such that $\tau \leq \binom{g+b-1}{g-1}$. Consider a τ-restricted (g,b)-scheme with $n \geq g+b$ parties and domain of keys K, which are distributed with the uniform a-priory probability distribution. The cardinality of the domain of pieces of every party in the scheme is at least $|K|^e$, where*

$$e \triangleq \max\left\{\tau, \frac{b-1}{g}\tau^{1-1/(g-1)}\right\}\ .$$

That is, the size of the pieces is at least e times the size of the key. For $\tau = 1$, the lower bound is $|K|^{1+\lfloor b/(g-1)\rfloor}$. That is, the size of the pieces in one-time key distribution schemes is at least $1 + \lfloor b/(g-1)\rfloor$ times the size of the key.

Proof: If $(b-1)/g < \tau^{1/(g-1)}$ then $e = \tau$, and the lower bound follows from Lemma 6.7. Otherwise, we convert the τ-restricted (g,b) scheme into a one time $(2,c)$ scheme (where c is a function of b, g and τ) such that the key in the new scheme is taken from a domain of cardinality $|K|^\tau$. To complete the proof, we apply the lower bound of Lemma 6.9.

In this conversion, we start with a τ-restricted (g,b) key distribution scheme, denoted $\mathcal{U}$, with $g+b$ parties. Define $a \triangleq \left\lceil (g-1)\tau^{1/(g-1)} \right\rceil$, and $c \triangleq \lfloor (g+b-1)/a \rfloor - 1$. We construct a one time $(2,c)$ scheme, called $\mathcal{U}'$, with $\lfloor (g+b-1)/a \rfloor + 1 = c+2$ parties. The new scheme is defined as follows: The piece of party P_1 is $\mathcal{U}'_1(r) \triangleq \mathcal{U}_1(r)$ and for every $2 \leq i \leq \lfloor (g+b)/a \rfloor$ the piece of party P_i is $\mathcal{U}'_i(r) = \langle \mathcal{U}_{a(i-2)+2}(r), \ldots, \mathcal{U}_{a(i-1)+1}(r) \rangle$. That is, in the new scheme party P_1 holds the piece of party P_1 from the old scheme, and party P_i gets the pieces of a disjoint parties of the original scheme. The number of original pieces we use is $(a\ \lfloor (g+b-1)/a \rfloor + 1) \leq g+b$, so this construction is possible. This conversion is illustrated in Figure 6.1. We consider the conference $\{P_i, P_j\}$ in the new scheme where $i > j$ (it is

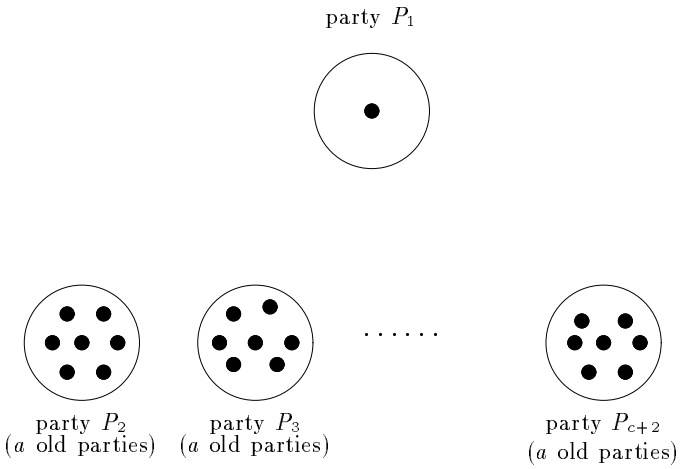

Figure 6.1: An illustration of the new scheme of Theorem 6.10
איור 6.1: תאור של הסכמה החדשה ממשפט 6.10

possible that $j = 1$). Since P_i holds a pieces of the original scheme and $a \geq (g-1)\tau^{1/(g-1)}$ there are

$$\binom{a}{g-1} \geq (a/(g-1))^{g-1} \geq \tau$$

conferences of g original parties, such that P_i holds $g-1$ of their pieces and P_j holds one piece. For every such conference, the parties P_i and P_j in the new scheme can generate the original key of the conference. That is, the conference $\{P_i, P_j\}$ in the new scheme can generate τ original keys. We view these keys as one new key taken from a domain of cardinality $|K|^\tau$.

We claim that this new key is distributed uniformly in the conditional space where pieces of c new parties (coalition members) are given. Denote the τ original conferences of size g by $G_1, G_2, \ldots, G_\tau$, and the generated keys by $k_1, k_2, \ldots, k_\tau$ respectively. Consider a coalition C of c new parties (not including P_i or P_j). If the coalition C has gained information on the concatenation of these keys then there exists an index i such that the pieces of C, the τ conversations, and the keys $k_1, \ldots, k_{i-1}$ reveal some information on the key k_i in the original scheme. This in turn, implies that in the original scheme there are pieces to all the parties except G_i that, together with the communications, reveal some information on the key k_i. But this coalition contains b original parties, which contradicts the τ-restricted security of $\mathcal{U}$, the original scheme.

Therefore, the scheme $\mathcal{U}'$ is a one time $(2, \lfloor (b+g-1)/a \rfloor - 1)$ scheme in which the key is distributed uniformly over a domain of cardinality $|K|^\tau$. Applying the lower bound of Lemma 6.9 to P_1, we conclude that U_1 – the domain of pieces of P_1 (in the new and original

scheme) – satisfies

$$|U_1| \geq |K|^{\tau(\lfloor (b+g-1)/a \rfloor)} > |K|^{\tau^{1-1/(g-1)}(b-1)/g} .$$

If $\tau = 1$ then $a = g-1$. Thus, $|U_1| \geq |K|^{\lfloor (b+g-1)/(g-1) \rfloor} = |K|^{1+\lfloor b/(g-1) \rfloor}$. □

We remark that the lower bound of Lemma 6.7 applies to weak τ-restricted schemes as well. On the other hand, the lower bound of Theorem 6.10 holds only for uniform a-priori distribution on the keys. For schemes where the keys are distributed with an arbitrary a-priori probability distributions, the lower bound is $2^{H(K)\cdot(1+\lfloor b/(g-1) \rfloor)}$. It is an open problem whether the lower bound of $|K|^{1+\lfloor b/(g-1) \rfloor}$ holds for weak one time schemes as well. Since Theorem 6.10 is proven by reduction to Lemma 6.9 then proving a $|K|^{b+1}$ lower bound for weak one time $(2, b)$ schemes suffices for proving the lower bound for every g.

6.5 Upper Bounds for Restricted Schemes

Blundo et. al. [29] present a one-time (g, b)-scheme in which the domain of pieces of each party is of cardinality $|K|^{g+b-1}$. (In [29] it is not pointed out that this scheme is only one-time secure). We improve their one-time scheme, and present a one-time key distribution scheme in which the domain of pieces of each party is of cardinality $|K|^{2+2(b-1)/g}$. This scheme is much more efficient than the unrestricted scheme. For example, for $g = b = n/2$, the cardinality of the domain of pieces in our scheme is $|K|^4$, regardless of n. Recall that for unrestricted schemes with these parameters, the cardinality of the domain of pieces is $|K|^{2^{\Omega(n)}}$ (Theorem 6.6).

We next describe our one-time scheme.

Lemma 6.11: *Let K be a domain of keys of cardinality q^g, such that q is a prime-power which is greater or equal to $\sqrt{n}$. There exists a one-time (g, b) scheme with n parties and domain of keys K (with uniform a-priori probability distribution) in which the cardinality of the domain of pieces of every party is $|K|^{2+2\cdot(b-1)/g}$.*

Proof: For clarity of the exposition, we first assume that $n = g + b$. In this case let $|K| = q^g$ for some positive number $q \geq 2$ (not necessary a prime power). For every pair of parties the dealer chooses two independent random strings from $\{0, \ldots, q-1\}$ denoted $k'_{i,j}, k''_{i,j}$. The piece of each party is the $2(n-1)$ random strings corresponding to the pairs that contain the party. When the parties of a conference G want to generate a conference key, every party $P_i \in G$ picks at random $k_i \in \{0, \ldots, q-1\}$. The conference key k of the set G is the concatenation of these random k_i's, namely $k \triangleq k_1 \circ k_2 \circ \ldots \circ k_g$. Every party $P_i \in G$ should send k_i to every $P_j \in G$. Party P_i can send messages only on a public channel, so, he uses the common random string common to P_i and P_j as a one time pad. Since party P_i has to send the sub-secret k_i to P_j and P_j has to send k_j to P_i, they have to use different random strings. That is, P_i sends $(k_i + k'_{i,j}) \bmod q$ if $i < j$, and $(k_i + k''_{i,j}) \bmod q$ if $i > j$. Therefore, every random string is used only once. Hence, the messages that are sent are

all uniformly distributed and independent of the conference key of G and the pieces of any coalition B with b parties (provided $G \cap B = \emptyset$). This implies that the scheme is secure.

We now describe our communicating one-time (g, b)-scheme for every $n \geq g + b$. The scheme is similar to the case $n = g + b$, with one change. To guarantee that the scheme is secure, we make sure that from the point of view of every coalition of size at most b the pads used in the generation of a key are uniformly distributed and independent of each other. On the other hand, we do not require that all $2^{\binom{n}{2}}$ strings that are given to the parties are totally independent. It suffices that the dealer deals vectors of pieces according to the non-communicating $(2, g + b - 2)$ scheme of Blom [24] for n parties, with keys taken from a domain of cardinality $|K|^{2/g} = q^2$. The pads used in the generation of a key are the keys of this scheme. Before we discuss the security of the scheme, we consider its efficiency. In Blom's scheme the cardinality of the domain of keys has to be at least n – the number of parties. Hence, we require that $q^2 \geq n$. Furthermore, in Blom's scheme the domain is a finite field, so q has to be a prime-power. We use the non-communicating $(2, g + b - 2)$-scheme with domain of keys of cardinality $q^2 = |K|^{2/g}$. So the cardinality of the domain of pieces of each party is

$$(|K|^{2/g})^{g+b-1} = |K|^{2+2\cdot(b-1)/g} .$$

The security of the scheme follows Claim 6.2 in which we proved that the collection of keys of all pairs of parties in a set G of g parties in a non-communicating $(2, g + b - 2)$ scheme is uniformly distributed and independent of the pieces of a coalition B of b parties. Since these keys are used as one-time pads, it follows that no information is leaked on the generated key, and therefore the scheme is secure. The formal proof that Claim 6.2 implies the security property of the one time scheme appears in Claim 6.13 in Section 6.5.1. □

Our scheme is non-interactive; that is, the messages of each member of G does not depend on other messages. Another property of our scheme is that for a fixed b, the cardinality of the domain of pieces of each party is a monotonically *decreasing* function of g. This feature stands in contrast to unrestricted (g, b) schemes, where the cardinality of the domain of pieces of each party is a monotonically *increasing* function of g. Recently [33], slightly improved our schemes. For example, they show for $b = 1$ a one-time scheme in which the size of the pieces is $(1 + 2/g)$ times the size of the keys (compared to 2 times size of pieces in our scheme).

In the proof of Lemma 6.11, the conference key of G (i.e. $k = k_1 \circ k_2 \circ \ldots \circ k_j$) is distributed uniformly in K. It is possible to change this a-priori probability distribution on the keys. One way to achieve this goal is to first generate a key k as in the previous way. Then an arbitrary party chooses the real key k' for the conference according to any desired distribution. The party sends the message $(k + k') \bmod q$ to all the parties in G.

We remark that the one time scheme cannot be reused. For example, if parties $\{P_1, P_2\}$ are members of two conferences G_1, G_2 then the part of the keys generated by them in the two conferences will be known to all the parties in $G_1 \cup G_2$. To extend our scheme to a τ-secure one, we use τ independent copies of the one time scheme. The generation of a key for each conference is done using a different copy of the one time scheme. Since the

copies are independent, each conference does not add any information to other conferences. Hence, the security of the one-time scheme implies the security of the τ-restricted scheme. We summarize this construction in the next theorem.

Theorem 6.12: *Let K be a domain of keys, such that $|K| = q^g$ for some prime-power $q \geq \sqrt{n}$. There exists a τ-restricted (g,b) key distribution scheme with n parties and domain of keys K (with any a-priori probability distribution), in which the domain of pieces of each party has cardinality $|K|^{2\tau(1+(b-1)/g)}$. That is, the size of the pieces in the scheme is $2\tau(1+(b-1)/g)$ times the size of the key.*

This τ-restricted communicating schemes requires that the parties hold a counter, which is incremented each time a conference key is generated. Given such a reliable counter, *active attack* by parties sending messages deviating from the protocol, do not reveal information on different conferences. Such attack could only prevent the generation of the present conference key. Our scheme does not work in the absence of a reliable counter. It is an open question to construct an efficient τ-restricted scheme (for $\tau > 1$) in which the keys and messages do not depend on a counter or any other history of previous conferences.

6.5.1 Formal Proof of Security of the One-time Scheme

In this section, we prove that Claim 6.2 implies the security requirement of our one-time key distribution scheme.

Claim 6.13: *The one-time (g,b) scheme presented in the proof of Lemma 6.11 has the 1-restricted security property.*

Proof: Since in our scheme the generated key is $\langle r_1, r_2, \ldots, r_g \rangle$ which is only a function of the local random inputs of the parties (and not their pieces), we denote this key by $k_G(\vec{r}_G)$. We fix vector of pieces $\langle u_1, \ldots, u_n \rangle$ which is dealt with positive probability, and a conference G of g parties. Furthermore, we denote by EV the event that $\bigwedge_{\ell \in B}(\mathcal{U}_\ell(r) = u_\ell)$. By Bayes rule, it is enough to prove that for every coalition B of b parties, such that $G \cap B = \emptyset$, every possible key $k = k_1 \circ k_2 \circ \ldots \circ k_g \in K$, and every possible consistent messages $\{m_{i,j} : P_i, P_j \in G\}$ (where $m_{i,j}$ is the message sent by party P_i to help party P_j)

$$\Pr[\, k_G(\vec{r}_G) = k_1 \circ \ldots \circ k_g \,|\, \mathrm{EV} \wedge \bigwedge_{i<j} r_i + k'_{i,j} = m_{i,j} \wedge \bigwedge_{i>j} r_j + k''_{i,j} = m_{j,i} \,] = \frac{1}{|K|} \;, \qquad (6.4)$$

where the probability is taken over r – the random input of the dealer and $\vec{r}_G = \langle r_i : P_i \in G \rangle$ – the random inputs of the parties of the conference G. Let K' be a domain of keys of cardinality $|K|^{2/g}$ (the domain of keys of the non-communicating scheme). We first claim

that for every fixed key $k_G = \langle r_1, r_2, \ldots, r_g \rangle \in K$ it holds that

$$\begin{aligned}
&\Pr[\,\mathrm{EV} \wedge \bigwedge_{i<j} r_i + k'_{i,j} = m_{i,j} \wedge \bigwedge_{i>j} r_j + k''_{i,j} = m_{j,i}\,] \\
&\qquad = \Pr[\bigwedge_{P_i,P_j \in G} k_{i,j} = (m_{i,j} - r_i) \circ (m_{j,i} - r_j) \,|\, \mathrm{EV}\,] \cdot \Pr[\,\mathrm{EV}\,] \\
&\qquad = \frac{\Pr[\,\mathrm{EV}\,]}{|K'|^t} \,. \qquad (6.5)
\end{aligned}$$

Equation (6.5) follows from Claim 6.2, and the fact that the keys of the non-communicating scheme are distributed uniformly (recall that $t = \binom{g}{2}$). We return to prove Equation (6.4).

$$\begin{aligned}
&\Pr[\,k_G(\vec{r}_G) = k \,|\, \mathrm{EV} \wedge \bigwedge_{i<j} r_i + k'_{i,j} = m_{i,j} \wedge \bigwedge_{i>j} r_j + k''_{i,j} = m_{j,i}\,] \\
&= \frac{\Pr[\,k_G(\vec{r}_G) = k \wedge \mathrm{EV} \wedge \bigwedge_{i<j} r_i + k'_{i,j} = m_{i,j} \wedge \bigwedge_{i>j} r_j + k''_{i,j} = m_{j,i}\,]}{\Pr[\,\mathrm{EV} \wedge \bigwedge_{i<j} k_i + k'_{i,j} = m_{i,j} \wedge \bigwedge_{i>j} k_j + k''_{i,j} = m_{j,i}\,]} \\
&= \frac{\Pr[\,\mathrm{EV} \wedge \bigwedge_{i<j} r_i + k'_{i,j} = m_{i,j} \wedge \bigwedge_{i>j} r_j + k''_{i,j} = m_{j,i}\,]}{\sum_{\vec{r}_G \in K} \Pr[\,\mathrm{EV} \wedge \bigwedge_{i<j} r_i + k'_{i,j} = m_{i,j} \wedge \bigwedge_{i>j} r_j + k''_{i,j} = m_{j,i}\,]} \\
&= \frac{1/|K'|^t}{|K| \cdot 1/|K'|^t} = \frac{1}{|K|} \,.
\end{aligned}$$

□

6.6 Comparison with the Computational Model

We contrast the results about key distribution schemes in the information theoretic model with those in the computational model, where parties are restricted to probabilistic polynomial time computations. Diffie and Hellman [51], in their pioneering work on public key cryptography, introduced a communicating scheme of key generation for conferences of size two. Their communicating scheme requires no dealer and no pieces. In this scheme a given communication *uniquely* determines the key, but it is (presumably) intractable for a third party to compute the key from the communication (of course, in our setting this information enables other parties to find the conference key). On the other hand, even in the computational model, a non-communicating scheme requires pieces taken from a domain which is at least as large as the domain of keys. So in the computational model, communication does reduce the size of pieces, up to complete elimination. As we said, in our setting communication does not reduce the size of pieces distributed to the parties by the server in unrestricted schemes. Fiat and Naor [54] present a non-communicating $(n, 1)$-scheme in the computational model. In their scheme, which is based on the assumed intractability of extracting root modulo composites (RSA), the domain of pieces has the same cardinality as the domain

of keys, $|K|$. Recall that in the computationally unbounded model a non communicating $(n, 1)$-scheme requires domain of pieces of size at least $|K|^n$ (i.e. a piece has the length of n keys).

Chapter 7

Secret Sharing with Public Reconstruction

All known constructions of information theoretic t-out-of-n secret sharing schemes require *secure, private* communication channels among the parties for the reconstruction of the secret. In this chapter we investigate the cost of performing the reconstruction over *public* communication channels. A naive implementation of this task distributes $O(n)$ one times pads to each party. This results in pieces whose size is $O(n)$ times the secret size. We present three implementations of such schemes that are substantially more efficient:

- A scheme enabling multiple reconstructions of the secret by different subsets of parties, with factor $O(n/t)$ increase in the pieces' size.
- A one-time scheme, enabling a single reconstruction of the secret, with $O(\log(n/t))$ increase in the pieces' size.
- A one-time scheme, enabling a single reconstruction by a set of size *exactly* t, with factor $O(1)$ increase in the pieces' size.

We prove that the first implementation is optimal (up to constant factors) by showing a tight $\Omega(n/t)$ lower bound for the increase in the pieces' size.

The rest of this chapter is organized as follows: In Section 7.1 we define secret sharing schemes with public reconstruction. In Section 7.2 we describe the unrestricted schemes, and in Section 7.3 the one time schemes. In Section 7.4 we introduce non-interactive, unrestricted schemes. In Section 7.5 we provide the lower bound for unrestricted schemes. Finally, in Section 7.6 we summarize our results and give two numerical examples of the sizes of pieces in our various schemes.

7.1 Definitions

In this section we define secret sharing scheme with public reconstruction (in contrast to traditional schemes with private channels as defined in Definition 3.6). In the following definition we consider only threshold schemes.

Definition 7.1 [Secret Sharing Schemes with Public Reconstruction]: *A* t-out-of-n secret sharing scheme with public reconstruction *is a secret sharing scheme satisfying the following two conditions:*

Reconstruction requirement *Any set of parties whose size is at least* t *can reconstruct the value of the secret after communicating via public channels. Every party in the reconstructing set gets the value of the secret with certainty (that is, the probability of errors is zero).*

Security requirement *Every disjoint coalition* B *of size at most* $t-1$ *has no information on the secret as defined in Definition 3.4. There are two variants we consider:*

1. Unrestricted schemes *in which a disjoint coalition* B *can overhear all communications taking place. The security is guaranteed even if several sets (maybe even all) reconstruct the secret using the public channel. In this case the view of a disjoint coalition is its pieces and all the communications that took place.*

2. One-time schemes *in which the security is guaranteed only if one set will reconstruct the secret. It is not known during system initialization which set will reconstruct the secret, and the dealer has to accommodate any possible set. In this case the view of a disjoint coalition is its pieces and the communication of one reconstructing set.*

The security should hold for any coalition of at most $t-1$ *parties. A special case is* $B = \emptyset$*, namely a listener who overhears all communications but has no pieces should gain no partial information about the secret.*

In traditional secret sharing schemes, while one set reconstructs the secret, no information is leaked to disjoint coalitions (due to the security of the channels). Hence, secure traditional schemes are always unrestricted. Furthermore, in traditional schemes, if a set can reconstruct the secret, then every superset of the set can reconstruct the secret. However, one-time secret sharing schemes with public reconstruction do not necessarily have this monotone property. To satisfy monotonicity, it is required that every party of the superset should know the reconstructed secret. However, it is not necessarily possible to "distribute" the secret to members of a superset without leaking information to other parties.

7.2 Unrestricted Schemes

In this section we construct unrestricted secret sharing schemes with public reconstruction, in which the size of the piece of every party is $O(n/t)$ times the size of the secret. We first

describe a simple scheme in which the size of the pieces is $O(n)$ times the size of the secret. Our $O(n/t)$ construction can be viewed as an optimization of this simple scheme. In this scheme, the dealer shares the secret using Shamir's secret sharing scheme [101]. The dealer also deals to every pair of parties two random strings whose size is the same as the size of the secret. These two random strings, which we call keys, are given to the two parties of the pair, and will be used as one-time pads. Overall, every party receives $2(n-1)$ keys, each one with the same size as the secret. When the parties in a set of size at least t wish to reconstruct the secret, all the parties "send" their pieces to the "leader" of the set, say the party with minimal index in the set. The leader gets at least t pieces (including his own), which enable him to reconstruct the secret. Then, the leader "sends" the secret to the other parties. The parties use their keys as one time pads to simulate private channels. Specificly, let P_{i_0} be the party with smallest identity in the set. Every party P_i, holding the piece s_i from Shamir's scheme, adds s_i and the first key of the pair $\langle P_{i_0}, P_i \rangle$ and sends this sum over the public channel (this is an addition in the appropriate finite field). The party P_{i_0} can reconstruct all the pieces from these messages, and therefore reconstruct the secret. Now, P_{i_0} sends messages, one message to every party in the reconstruction set. For every party P_i, he adds the secret and the second key of the pair $\langle P_{i_0}, P_i \rangle$ and sends this sum over the public channel. Since the one-time pads are independent, coalitions of parties disjoint to the reconstructing set do not gain any information on the pieces or the secret. Furthermore, even if many reconstructions take place, this does not leak any information to a disjoint set.

Suppose P_{i_0} is the leader in a set of size at least t. In the previous scheme, during the reconstruction for this set, only the keys that were given to P_{i_0} were used. To improve the space efficiency we will use all the keys of the parties in the reconstructing set. Like the one-time key distribution scheme of Lemma 6.11, there would be t leaders. We partition the secret into t sub-secrets, and share each sub-secret using Shamir's scheme. Now we choose t parties of the reconstructing set, and each one will be responsible for reconstructing one sub-secret. Each party will act as the leader in the previous scheme. That is, every leader receives pieces of his sub-secret from the other $t-1$ leaders (this is enough), but sends his sub-secret (after reconstruction) to every member of the reconstructing set. I.e., the size of the one time pads is only the size of the sub-secret. This way we can handle t sub-secrets "at the price of one". The domain of the secrets in the scheme is $\mathrm{GF}(q)^t$, where q is a prime-power such that $q > n$. (We require that $q > n$ since this is the requirement in Shamir's scheme.) In the scheme we view the secret as t sub-secrets from $\mathrm{GF}(q)$. The scheme is presented in Fig. 7.1.

As described, the scheme has two technical points which should be clarified. The first is the fact that in one reconstruction two parties P_i and P_j might need to exchange 4 different messages: Assume that P_i is responsible for reconstructing the sub-secret $s_{i'}$, and P_j is responsible for reconstructing the sub-secret $s_{j'}$. The party P_i has to receive from P_j the piece corresponding to the sub-secret $s_{i'}$, and then will send the sub-secret $s_{i'}$. Similarly, P_j has to receive from P_i the piece corresponding to the sub-secret $s_{j'}$, and then will send the sub-secret $s_{j'}$. This is the reason for giving them 4 common keys. The second difficulty is

Unrestricted Secret Sharing Scheme

Distribution stage:

Input: t sub-secrets $s_1, s_2, \ldots, s_t \in \mathrm{GF}(q)$

Pieces: Share each sub-secret s_i using Shamir's scheme for every i, where $1 \leq i \leq t$.
Denote the n pieces of sub-secret s_i by $s_{i,1}, s_{i,2}, \ldots, s_{i,n}$.
For every pair of parties generate 4 independent keys from $\mathrm{GF}(q)$.
Denote the keys of $\langle P_i, P_j \rangle$ by $k^1_{i,j}, k^2_{i,j}, k^3_{i,j}, k^4_{i,j}$.
The piece of party P_i is $s_{1,i}, \ldots, s_{t,i}$ and keys $k^1_{i,j}, k^2_{i,j}, k^3_{i,j}, k^4_{i,j}$ for $1 \leq j \leq n$.

Reconstruction stage:

A set $G = \{P_{i_1}, \ldots, P_{i_m}\}$ that wants to reconstruct the secret ($m \geq t$).
Every party in G announces whether he has previously reconstructed the secret.
Let P_{i_j} for $1 \leq j \leq t$ be the leaders of G.
Each leader P_{i_j} $(1 \leq j \leq t)$ sends (at most) $t-1$ messages to all other leaders that have not previously reconstructed the secret:

$s_{j',i_j} + k^1_{i_j,i_{j'}}$ to $P_{i_{j'}}$ for $1 \leq j' < j$

$s_{j',i_j} + k^2_{i_j,i_{j'}}$ to $P_{i_{j'}}$ for $j < j' \leq t$

Each leader P_{i_j} $(1 \leq j \leq t)$ computes s_j from $s_{j,i_1}, \ldots, s_{j,i_t}$.
Each leader P_{i_j} sends a message to every $P_{i_{j'}} \in G$ that has not previously reconstructed the secret:

$s_j + k^3_{i_j,i_{j'}}$ to $P_{i_{j'}}$ for $1 \leq j' < j$

$s_j + k^4_{i_j,i_{j'}}$ to $P_{i_{j'}}$ for $j < j' \leq m$

Each party of G concatenates the sub-secrets $s_1, s_2, \ldots, s_t$ to obtain the secret s.

Figure 7.1: Unrestricted t-out-of-n secret sharing scheme with public reconstruction

איור 7.1: סכמה לא מוגבלת לחלוקת סוד עם שחזור פומבי עבור סף t מתוך n

that in different reconstructions the same party can be responsible for different sub-secrets. This means that P_i will have to send to P_j two different messages, using the same key as a one time pad. This might leak information to disjoint coalitions. To overcome the problem, every party that participates in one reconstruction will remember the secret, and in latter reconstructions will inform other parties (in the clear) that he knows the secret. In such case, other parties will not send him any messages. He will continue to send the messages that he has to send according to the scheme (to "new" parties). Thus, every key is used as a one-time pad at most once (in the first reconstruction that the pair participates together). Therefore, the scheme satisfies the unrestricted security requirement.

Let us calculate the size of the piece of every party in this unrestricted scheme. Each party is given t pieces generated by Shamir's scheme for secrets taken from $\mathrm{GF}(q)$. The dealer also distributes to each party $4(n-1)$ keys taken from $\mathrm{GF}(q)$. Hence, each piece contains $(4n+t-4)$ elements from $\mathrm{GF}(q)$, compared to t elements from $\mathrm{GF}(q)$ for the secret. We summarize these results in the next theorem.

Theorem 7.2: *Let q be a prime-power such that $q > n$. The scheme of Fig. 7.1 is an unrestricted t-out-of-n secret sharing scheme with public reconstruction for secrets taken from $\mathrm{GF}(q)^t$. The piece of each party is an element of $\mathrm{GF}(q)^{4n+t-4}$. So the size of each piece is $1+4(n-1)/t$ times the size of the secrets.*

7.3 One-Time Schemes

In the unrestricted scheme, we need totally independent keys in order to guarantee the security of the scheme during repeated reconstructions. In this section we deal with the scenario where the secret is going to be reconstructed only once. For example, to enable the firing of a ballistic missile or opening of a sealed safe. In this case, total independence among the keys is not needed, and weaker independence requirements suffice. Pieces can therefore be taken from a smaller sample space, which translates into smaller size pieces. Specifically, we use Blom's key distribution scheme [24] for this purpose.

The first scheme we present enables one-time reconstruction of the secret by sets of size *exactly* t. The size of the pieces is a constant (less than 10) times the size of the secret, namely only $O(1)$ increase in pieces' size. We employ this "exactly t" scheme as a building block for "at least t" schemes. We use $1+\log(n/t)$ independent instances of "exact schemes" for thresholds $t, 2t, 4t, \ldots$ up to n, and an additional instance of size t. Now, given any set G with m parties ($m \geq t$), we represent it as a union of subsets (not necessary disjoint) with cardinalities $t, 2t, 4t, \ldots$ – at most two subsets of cardinality t and at most one subset of cardinality $2^i t$ for each $i \geq 1$. The secret is now separately reconstructed by each subset. Any member of G takes part in at least one of these reconstructions, and thus learns the secret. On the other hand, any disjoint coalition containing at most $t-1$ parties gets no partial information on the secret from any single instance. Due to the independence of the instances, this remains valid with respect to the joint reconstructions. We get a one-time

scheme for set of size at least t, with just $O(\log(n/t))$ increase in piece size. We now describe in detail the "exactly t" scheme. The distribution phase is depicted in Fig. 7.2.

Distribution in Exactly t-out-of-n one-time scheme

Input: secret $s \in \mathrm{GF}(q)^t$.

Consider the secret as t secrets $s_1, \ldots, s_t \in \mathrm{GF}(q)$.
Share each secret s_i using Shamir's t-out-of-n secret sharing scheme.
Let $b = \min\{2t-3, n-2\}$.
Generate pieces using $(2,b)$-key distribution scheme with key domain $\mathrm{GF}(q)^4$
(which we consider as 4 keys from $\mathrm{GF}(q)$).

Piece of P_j: the j-th piece of each s_i, and the piece of the key distribution scheme.

Figure 7.2: One-time exactly t-out-of-n secret sharing scheme with public reconstruction
איור 7.2: סכמה חד פעמית לחלוקת סוד עם שחזור פומבי עבור בדיוק t מתוך n

The reconstruction is done exactly as in the unrestricted scheme. The security of one reconstruction of a set of exactly t parties follows from the property of $(2, 2t-3)$ key distribution schemes proved in Claim 6.2: Given the pieces of any disjoint coalition of at most $t-1$ parties, the keys held by any pair of parties in a set of size t are distributed uniformly and independently. Thus, when used as one-time pads, the reconstruction is secure (using the same arguments as in the unrestricted case). This scheme uses t pieces of Shamir's t-out-of-n secret sharing scheme with secrets taken from $\mathrm{GF}(q)$. In addition, each party gets a piece of a $(2, 2t-3)$ key distribution scheme with keys taken from $\mathrm{GF}(q)^4$ – these pieces are taken from $\mathrm{GF}(q)^{4(2t-2)}$. Overall, the total piece contains $(9t-8)$ elements from $\mathrm{GF}(q)$ (if $2t > n+1$, then the pieces are even shorter). Recall that the secret is taken from $\mathrm{GF}(q)^t$, and therefore the size of the piece is less than 9 times the size of the secrets.

In this scheme, the domain of secrets has to be $\mathrm{GF}(q)^t$ (for some prime-power q). Restricting the domain of the secrets to such cardinality can cause problems when we employ simultaneously many schemes with the same secret but with different thresholds. To overcome this, given any domain of secrets, we consider a slightly bigger domain whose size (which can depend on the threshold) is of the desired form. That is, given a secret of size ℓ which is at least $t \log n$, we choose a prime power q such that $\ell \leq t \log q$, and use the previous scheme with secrets of size $\ell' = t \log q$. Choosing $q = 2^{\lceil \ell/t \rceil}$, we have $\ell' = t \lceil \ell/t \rceil \leq \ell + t$. If we assume that $\ell > 9t$ then the size of the piece is $9\ell' \leq 9(\ell + t) \leq 10\ell$.

Theorem 7.3: *Let ℓ be a natural number such that $\ell > \max\{t \log n, 9t\}$. There exists a one-time exactly t-out-of-n secret sharing scheme with public reconstruction in which the size of the secret equals ℓ, and the size of the piece of each party is less than 10 times the size of the secrets.*

One-time schemes are a special case of traditional secret sharing schemes even if only sets of exactly t can securely reconstruct the secret, since every set of at least t parties has enough information to reconstruct the secret on secure private channels. Thus, the size of each piece has to be at least the size of the secret [69]. Therefore, our scheme is tight up to a constant factor. We can slightly improve this lower bound, by observing that every one-time exactly t-out-of-n secret sharing scheme with public reconstruction can be used as a one-time communicating $(t, t-1)$ key distribution scheme (for $t \leq n/2$). By Theorem 6.10, the size of the piece in every one-time $(t, t-1)$ key distribution scheme is at least twice the size of the key. Therefore, the size of the piece in every one-time secret sharing scheme with public reconstruction is at least twice the size of the secret.

In Fig. 7.3 we describe the one time scheme in which every set of *at least* t parties can securely reconstruct the secret.

One-time Secret Sharing Scheme

Distribution stage:
Input: secret s of size ℓ
Share the secret s using two independent copies of a
    one time exactly t-out-of-n secret sharing schemes.
For every i, $1 \leq i < \log(n/t)$:
    Share s with an exactly $2^i t$-out-of-n one time secret sharing scheme.

Reconstruction stage:
A set $G = \{P_{i_1}, \ldots, P_{i_m}\}$ that wants to reconstruct the secret ($m \geq t$).
Cover the set G by (possibly intersecting) sets of size $2^i t$
    (at most one set for every $i > 1$, and at most 2 sets of size t).
Each set of size $2^i t$ independently reconstructs the secret using
    the pieces of the exactly $2^i t$-out-of-n secret sharing scheme.

Figure 7.3: One-time t-out-of-n secret sharing scheme with public reconstruction
איור 7.3: סכמה חד פעמית לחלוקת סוד עם שחזור פומבי עבור סף t מתוך n

Theorem 7.4: *The scheme of Fig. 7.3 is a one-time t-out-of-n secret sharing scheme with public reconstruction in which every set of parties of size at least t can securely reconstruct the secret. If the size of the secrets ℓ is larger than $\max\{n\log n, 9n\}$, then the size of the pieces of every party is less than $10(\log(n/t)+2)$ times the size of the secrets.*

Remark If we require that the size of the secret ℓ is greater than $n^2\log n$, then we can construct a scheme in which the size of the pieces is only $2\log(n/t)+O(1)$ times the size of the secret, i.e. a smaller leading constant. To achieve this goal we use a slightly weaker building block (instead of the exactly t-out-of-n scheme). This building block is a scheme in which exactly $2^i t$ parties can reconstruct the secret, while any coalition of size $t-1$ does not gain any information on the reconstructed secret. Schemes which satisfy this requirement on secure, private channels were presented by Blakley and Meadows [23] (see also [59, 69, 79]), and are called ramp schemes. We use ramp schemes instead of regular secret sharing schemes to design our building boxes. Assume that sets of m parties should be able to reconstruct the secret, while sets of fewer than t parties should get no information about the secret. The size of the piece in such ramp scheme is $1/(m-t+1)$ times the size of the secret, i.e. substantially smaller than traditional secret sharing schemes.

For the scheme with public reconstruction, assume that the domain of secrets is $\mathrm{GF}(q)^{m(m-t+1)}$. That is, the dealer has m secrets, each one taken from $\mathrm{GF}(q)^{m-t+1}$. The piece of each party is one piece of the ramp scheme for m sub-secrets, each piece taken from $\mathrm{GF}(q)$. In the reconstruction of the secret by a subset containing exactly m parties, each party will be responsible for one sub-secret. Each pair of parties in this set first exchanges two pieces of the ramp scheme. Now each party reconstructs his sub-secret, and every pair of parties exchange two sub-secrets. Therefore, every pair of parties needs two keys of a $(2, m+t-3)$-key distribution scheme with keys from $\mathrm{GF}(q)$ (the domain of pieces of the ramp scheme), as well as two keys from a $(2, m+t-3)$-key distribution scheme with keys from $\mathrm{GF}(q)^{m-t+1}$ (the domain of sub-secrets). Overall, the piece of each party is an element taken from $\mathrm{GF}(q)^{3m+2t-4+2(m+t-2)(m-t+1)}$. That is, the size of the piece is $2+2t/m+O(1/(m-t))$ times the size of the secrets. In the t-out-of-n scheme for every set, we use these schemes with $m = t, 2t, 4t, \ldots$ – therefore the size of the piece is only $2\log(n/t)+O(1)$ times the size of the secret. In this construction we required that the size of the secret m is greater than $n^2\log n$, this requirement can be weakened to $m \geq n\log^3 n$.

7.4 Unrestricted Non-Interactive Schemes

A secret sharing scheme with public reconstruction is called *non-interactive* if the messages sent by each party depend only on his piece (and not on messages received during the reconstruction). Non-interactive schemes are simpler to implement, as they require less synchronization. Therefore, they are desirable from practical point of view. In this section we present non-interactive, unrestricted t-out-of-n schemes. The size of the pieces in these

schemes is n times the size of the secret. This represents a slight improvements (by a factor of 2) over the interactive scheme of Section 7.2 for $t = 2$, but is strictly less efficient (in terms of piece size) for $t \geq 5$. We extend these schemes to general access structures. The size of the piece in our public reconstruction schemes is n times the size of the piece in the original scheme. For general access structure it is typically not a significant increase, as the best schemes for most access structures to date require pieces whose size is exponential in n.

We first present a simple, non-interactive, 2-out-of-n secret sharing scheme. Let $s \in \mathcal{Z}_m$ be the secret which the dealer wants to share. The dealer chooses n independent random elements from $\mathcal{Z}_m$, denoted $r_1, \ldots, r_n$. The piece of P_i is $r_1, \ldots, r_{i-1}, r_i + s, r_{i+1}, \ldots, r_n$. Each piece is uniformly distributed in $\mathcal{Z}_m^n$, regardless of the secret. Hence, prior to any reconstruction every party has no information on the secret (as defined in Definition 3.4). To reconstruct the secret, P_i sends the message r_j, and P_j sends the message r_i. Now, P_i, who holds $r_i + s$, hears the message r_i, so he can reconstruct the secret. Every third party hears messages that he already knows, and gains no information on the secret. That is, the reconstruction is secure. The size of the pieces in this scheme is n times the size of the secret. During the reconstruction in this scheme every party is deterministic and sends only one message that depends only on its piece.

The notion of secret sharing scheme with public reconstruction is naturally extended to every monotone access structure. The unrestricted, non-interactive, 2-out-of-n scheme can be generalized to an unrestricted, non-interactive scheme realizing an access structure $\mathcal{A}$. Assume there is a traditional secret sharing scheme realizing $\mathcal{A}$ with domain of secrets S and domain of pieces U. In our scheme we use the following observation: Denote by $\mathcal{A}_i$ the access structure $\mathcal{A}_i = \{B \ : \ B \cup \{P_i\} \in \mathcal{A}\}$. There exists a traditional secret sharing scheme realizing $\mathcal{A}_i$ in which the domain of pieces is U (fix some possible piece u for P_i, which would be public knowledge, and share the secret using the scheme for $\mathcal{A}$ conditioned on the fact that the piece of P_i is u).

We now describe an unrestricted, non-interactive scheme realizing $\mathcal{A}$ with domain of pieces $S \times U^{n-1}$. To share a secret s, the dealer chooses n random independent elements from S, the domain of secrets, denoted $r_1, \ldots, r_n$. For every i, the dealer distributes the piece $r_i + s$ to P_i, and shares r_i among $\{P_1, \ldots, P_{i-1}, P_{i+1}, \ldots, P_n\}$ using the scheme realizing $\mathcal{A}_i$ with domain of pieces U. That is, the piece of P_i is $r_i + s$ together with the pieces of the $n-1$ schemes realizing $\mathcal{A}_1, \ldots, \mathcal{A}_{i-1}, \mathcal{A}_{i+1}, \ldots, \mathcal{A}_n$ with secrets $r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_n$ respectively. The total piece is an element taken from $S \times U^{n-1}$. Now, when a subset B wishes to reconstruct the secret, every $P_i \in B$ sends (in the open) the piece of the secret r_j to every $P_j \in B$. Thus, P_i holds $r_i + s$ and hears the pieces of $B \setminus \{P_i\}$ from the scheme realizing $\mathcal{A}_i$ with the secret r_i. Since $B \setminus \{P_i\} \in \mathcal{A}_i$, the party P_i can reconstruct r_i, and with $r_i + s$ reconstructs the secret.

We next claim that the reconstructions are secure. That is, every coalition $C \notin \mathcal{A}$ hearing communications during the reconstruction of the secret by all sets in $\mathcal{A}$ that are disjoint to C does not gain information on the secret. The information that such coalition gains from the communication is at most the r_i's for $P_i \notin C$. On the other hand, $C \notin \mathcal{A}_i$ for any $P_i \in C$.

Therefore, the coalition C gets no information on such r_i, so even though he knows $r_i + s$, this gives no information on s. Since the $r_1, \ldots, r_n$ are random independent elements, C does not gain any information on the secret. That is, the reconstruction is secure. Thus,

Theorem 7.5: *Assume there exists a (traditional) secret sharing scheme realizing $\mathcal{A}$ with domain of secrets S and domain of pieces U. Then there exists an unrestricted, non-interactive secret sharing scheme realizing $\mathcal{A}$ with public reconstruction for secrets taken from S. The piece of each party is an element of $S \times U^{n-1}$. So the size of each piece is at most n times the size of the pieces in the original scheme.*

We can apply the previous construction for threshold schemes using Shamir's scheme.

Corollary 7.6: *Let q be a prime-power such that $q > n$. There exists an unrestricted, non-interactive t-out-of-n secret sharing scheme with public reconstruction for secrets taken from $\mathrm{GF}(q)$. The piece of each party is an element of $\mathrm{GF}(q)^n$. So the size of each piece is n times the size of the secret.*

7.5 Lower Bounds for Unrestricted Schemes

In this section we prove an $\Omega(n/t)$ lower bound on the increase in the pieces' size for unrestricted t-out-of-n schemes. The specific lower bound that we prove is tight for $t = 2$ (by the non-interactive scheme of Section 7.4). For $t > 2$ our lower bound is tight up to a constant factor (by the interactive scheme of Section 7.2). We first prove an $\Omega(n)$ lower bound on the increase in size of pieces for 2-out-of-n schemes. Then, we show that this lower bounds translates into an $\Omega(n/t)$ increase for t-out-of-n schemes.

We start with the lower bound for $t = 2$. The proof uses entropy and mutual information. For definitions of these information theoretic terms, the reader can refer to [42] and Appendix A. We assume an arbitrary probability distribution on the secrets, and we denote the secret by the random variable S.

The intuition behind the proof is that P_i has to expose $\mathrm{H}(S)$ *"new"* bits of his piece in *each* reconstruction, and P_i can participate in $n-1$ reconstructions. After all $n-1$ reconstructions, the uncertainty of the piece of P_i has to remain at least $\mathrm{H}(S)$, as an outsider who listened to all reconstructions still has $\mathrm{H}(S)$ uncertainty on the secret. Thus, the original entropy of the piece has to be at least $n \cdot \mathrm{H}(S)$.

Without loss of generality, we prove the claim for P_1. To prove the lower bound on P_1's piece, we only use the requirement that P_1 can reconstruct the secret together with every other P_j (we do not care if other pairs can or cannot reconstruct the secret). We start with some notation. Denote by S_i the piece given to P_i in the initial distribution phase, and by $C_{i,j}$ the messages exchanged between P_i and P_j while they reconstruct the secret (all these are random variables). We denote $C = C_{1,3} \ldots C_{1,n}$, the concatenation of all messages exchanged between P_1 and $P_3, \ldots, P_n$. Recall that the communication $C_{1,2}$, together with

P_2's piece S_2, enable P_2 to reconstruct the secret S. On the other hand, the communication C and S_2 give no information (to P_2) about the secret. These facts will imply the next claim.

Claim 7.7: $H(C_{1,2}|S_2C) \geq H(S)$.

Proof: Since P_2 can reconstruct the secret S, given his piece S_2 and the messages $C_{1,2}$ exchanged between P_1 and P_2, the conditional entropy $\mathrm{H}(S|C_{1,2}S_2)$ equals 0. On the other hand, P_2 gets no information about the secret S from his own piece S_2 and all messages C exchanged between P_1 and the other $n-2$ parties. Therefore the conditional entropy $\mathrm{H}(S|S_2C)$ equals $\mathrm{H}(S)$. Now, consider the conditional mutual information $\mathrm{I}(C_{1,2}; S|S_2C)$ of the message $C_{1,2}$ and the secret S, given the piece S_2 and C. We have

$$\begin{aligned} \mathrm{H}(C_{1,2}|S_2C) - \mathrm{H}(C_{1,2}|SS_2C) &= \mathrm{I}(C_{1,2}; S|S_2C) \\ &= \mathrm{H}(S|S_2C) - \mathrm{H}(S|C_{1,2}S_2C) \\ &= \mathrm{H}(S) \ . \end{aligned}$$

Since the entropy is non-negative, $\mathrm{H}(C_{1,2}|S_2C) \geq \mathrm{H}(S)$. □

The next claim is the heart of the proof of the lower bound. It states that the mutual information between S_1 and $C_{1,2}$ given the "other" communication C is at least $\mathrm{H}(S)$. Intuitively, since P_2 does not know the secret prior to the reconstruction, and knows it after the reconstruction, P_2 has to receive $\mathrm{H}(S)$ bits of information which could only originate in S_1 and passed through the communication $C_{1,2}$. Hence, $C_{1,2}$ must contain $\mathrm{H}(S)$ bits of information originating from the piece S_1. Claim 7.8 is stated for deterministic parties – the outgoing messages are determined by the given piece and previous incoming messages. An analogous statement is proved in Section 7.5.1 for randomized parties, whose outgoing messages could in addition depend on random local inputs.

Claim 7.8: *For deterministic reconstruction protocols we have*

$$I(C_{1,2}; S_1|C) = H(S_1|C) - H(S_1|C_{1,2}C) \geq H(S) \ .$$

Proof: Since P_1 and P_2 are deterministic, and their domain of pieces is finite, there is a bound k on the maximum number of communication rounds which can take place during the reconstruction of the secret. Denote by A_i the i-th message sent by P_1 to P_2, and similarly, let B_i be the i-th message sent by P_2 to P_1. Then, without loss of generality, $C_{1,2} = A_1B_1 \ldots A_kB_k$. The message A_i is determined by the piece S_1 and previous messages, that is, $\mathrm{H}(A_i|S_1A_1B_1 \ldots A_{i-1}B_{i-1}) = 0$. The following inequality holds for any deterministic communication protocol:

$$\begin{aligned} \mathrm{H}(C_{1,2}|S_1C) &= \mathrm{H}(A_1B_1 \ldots A_kB_k|S_1C) \\ &= \sum_{i=1}^{k} \mathrm{H}(A_i|S_1CA_1B_1 \ldots A_{i-1}B_{i-1}) + \sum_{i=1}^{k} \mathrm{H}(B_i|S_1CA_1B_1 \ldots A_{i-1}B_{i-1}A_i) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{k} \mathrm{H}(B_i | S_1 C A_1 B_1 \ldots A_{i-1} B_{i-1} A_i) \\
&\leq \sum_{i=1}^{k} \mathrm{H}(B_i | C A_1 B_1 \ldots A_{i-1} B_{i-1} A_i) \ .
\end{aligned}
$$

Similarly, $\mathrm{H}(C_{1,2}|S_2 C) \leq \sum_{i=1}^{k} \mathrm{H}(A_i | C A_1 B_1 \ldots A_{i-1} B_{i-1})$. Combing the two inequalities

$$
\begin{aligned}
\mathrm{H}(C_{1,2}|S_1 C) + \mathrm{H}(C_{1,2}|S_2 C) &\leq \sum_{i=1}^{k} (\mathrm{H}(B_i | C A_1 B_1 \ldots B_{i-1} A_i) + \mathrm{H}(A_i | C A_1 B_1 \ldots A_{i-1} B_{i-1})) \\
&= \mathrm{H}(A_1 B_1 \ldots A_k B_k | C) \\
&= \mathrm{H}(C_{1,2}|C) \ .
\end{aligned}
$$

This inequality, together with Claim 7.7, implies

$$
\begin{aligned}
\mathrm{I}(C_{1,2}; S_1 | C) &= \mathrm{H}(C_{1,2}|C) - \mathrm{H}(C_{1,2}|S_1 C) \\
&\geq \mathrm{H}(C_{1,2}|S_2 C) \\
&\geq \mathrm{H}(S)
\end{aligned}
$$

□

We are now ready to prove our lower bound for $t = 2$.

Claim 7.9: *In any unrestricted 2-out-of-n secret sharing scheme with public reconstruction, the piece of each participant, S_i, satisfies*

$$H(S_i) \geq n \cdot H(S) \ .$$

Proof: We first note that by Definition 7.1 a listener, who overhears all communication involving P_1, gets no information on the secret. Therefore,

$$\mathrm{H}(S | C_{1,2} C_{1,3} \ldots C_{1,n}) = \mathrm{H}(S) \ .$$

On the other hand, given P_1's piece, this communication determines the secret, so

$$\mathrm{H}(S | S_1 C_{1,2} C_{1,3} \ldots C_{1,n}) = 0 \ .$$

Therefore,

$$
\begin{aligned}
\mathrm{H}(S) &= \mathrm{H}(S | C_{1,2} C_{1,3} \ldots C_{1,n}) - \mathrm{H}(S | S_1 C_{1,2} C_{1,3} \ldots C_{1,n}) \\
&= \mathrm{I}(S; S_1 | C_{1,2} C_{1,3} \ldots C_{1,n}) \\
&= \mathrm{H}(S_1 | C_{1,2} C_{1,3} \ldots C_{1,n}) - \mathrm{H}(S_1 | S C_{1,2} C_{1,3} \ldots C_{1,n}) \ ,
\end{aligned}
$$

and in particular

$$\mathrm{H}(S_1 | C_{1,2} C_{1,3} \ldots C_{1,n}) \geq \mathrm{H}(S) \ .$$

Claim 7.8 (or Claim 7.13 for the case of randomized protocols) states that

$$\mathrm{H}(S_1|C_{1,3}\dots C_{1,n}) - \mathrm{H}(S_1|C_{1,2}C_{1,3}\dots C_{1,n}) \geq \mathrm{H}(S) \ .$$

Similarly it holds that

$$\begin{aligned} \mathrm{H}(S_1|C_{1,4}\dots C_{1,n}) - \mathrm{H}(S_1|C_{1,3}C_{1,4}\dots C_{1,n}) &\geq \mathrm{H}(S) \ . \\ &\vdots \\ \mathrm{H}(S_1) - \mathrm{H}(S_1|C_{1,n}) &\geq \mathrm{H}(S) \ . \end{aligned}$$

Summing these n inequalities, we conclude that

$$\mathrm{H}(S_1) \geq n \cdot \mathrm{H}(S) \ .$$

□

We next show that this lower bounds on increase in size of pieces for 2-out-of-n schemes translates into $\Omega(n/t)$ increase for t-out-of-n schemes.

Theorem 7.10: *In every unrestricted t-out-of-n secret sharing scheme with public reconstruction the size of the pieces of every party is at least $\lfloor 1 + (n-1)/(t-1) \rfloor$ times the size of the secrets.*

Proof: Consider any t-out-of-n scheme. Denote the party whose piece is shortest by P_1. We construct an unrestricted 2-out-of-($\lfloor 1 + (n-1)/(t-1) \rfloor$) scheme in which the entropy of S_1 – the piece of P_1 – is the same. Hence, by Claim 7.9 its entropy is at least $(\lfloor 1 + (n-1)/(t-1) \rfloor)\mathrm{H}(S)$. Since the scheme is secure whatever the distribution on the secrets is, we can assume uniform distribution on the secrets. In this case $\mathrm{H}(S) = \log |S|$, which is the size of the secret. Since $\mathrm{H}(S_1) \leq \log |S_1|$, the size of the piece of P_1 is at least $\lfloor 1 + (n-1)/(t-1) \rfloor$ times the size of the secrets.

The construction is simple: the dealer gives P_1 the piece of P_1 in the original scheme, and every other party gets pieces of $t-1$ disjoint parties. Since every party has at most $t-1$ pieces, he does not gain any information on the secret even after hearing communications. On the other hand, every 2 parties have at least t pieces, therefore they can communicate on a public channel, and securely reconstruct the secret. □

7.5.1 Lower Bound for Probabilistic Parties

In the proof of Claim 7.8 we assumed that the parties are deterministic during the reconstruction of the secret. In this section we prove the same claim without this assumption. Recall that S_1 is the piece of P_1, $C_{1,2}$ is the communication generated in the reconstruction of the secret by P_1 and P_2, and C is the communication in previous reconstructions. We prove that the mutual information between S_1 and $C_{1,2}$ given C is at least the entropy of

the secret, even if the parties may toss coins during the reconstruction. That is, party P_1 has an independent local random string, denoted R_1, and the messages he generates are a deterministic function of his piece, his random input, and previous messages. As the claim concerns the piece of P_1, we can assume that other parties in the system are deterministic (the dealer can supply a random string to the other parties as part of their pieces). Since R_1 is independent of the pieces and the secret, the mutual information between R_1 and the pieces and the secret is zero, i.e. $\mathrm{I}(R_1; SS_1S_2 \ldots S_n) = 0$. We next prove that the mutual information between R_1 and the pieces of other parties, given S_1 and a communication M is zero (where M is any prefix of $CC_{1,2}$). The claim can be proven directly by induction on the number of messages sent in M. We avoid this induction as we show that our claim can be formulated as a special case of Claim 6.8 (actually, Claim 6.8 is proven by induction).

Claim 7.11: *Let M be a prefix of the communication exchanged between the parties in the system. Then, $H(R_1 \mid SS_2S_1M) = H(R_1 \mid S_1M)$.*

Proof: Consider a scenario in which one party holds R_1, and a second party holds the secret S and all the pieces – $S_1, S_2, \ldots, S_n$. They communicate via a public channel and the first message is sent by the second party and equals S_1. From now on, the first player can generate the messages of P_1 and the second player can generate all other messages. Thus, the two parties can continue to communicate and generate M. By Claim 6.8 (proved in [77] and [2, Lemma 2.2]), communicating S_1 and M can only decrease the mutual information, that is,

$$\mathrm{I}(R_1; SS_1S_2 \ldots S_n | S_1M) \leq \mathrm{I}(R_1; SS_1S_2 \ldots S_n) = 0 \ .$$

Recall that, $\mathrm{I}(R_1; SS_1S_2 | S_1M) \leq \mathrm{I}(R_1; SS_1S_2 \ldots S_n | S_1M)$. Since the mutual information is non-negative,

$$\mathrm{H}(R_1|S_1M) - \mathrm{H}(R_1|SS_1S_2M) = \mathrm{I}(R_1; SS_1S_2|S_1M) = 0 \, .$$

□

We restrict our discussion to protocols with an absolute bound k on the number of rounds in each communication $C_{i,j}$. (The case where the protocol terminates after a finite number of rounds with probability 1 can be handled similarly). Denote by A_i the i-th message sent by P_1 to P_2, and similarly, B_i to be the i-th message sent by P_2 to P_1. I.e, $C_{1,2} = A_1B_1 \ldots A_kB_k$. We next prove that the dependence of A_i on S_1 and the previous messages is greater than the dependence of A_i on S_2, the secret S, and the previous messages. Formally,

Claim 7.12: $H(A_i|SS_2CA_1B_1 \ldots A_{i-1}B_{i-1}) \geq H(A_i|S_1CA_1B_1 \ldots A_{i-1}B_{i-1})$.

Proof: On one hand, since the entropy is non-negative

$$\begin{aligned}
&\mathrm{I}(A_i; R_1|SS_1S_2CA_1B_1 \ldots A_{i-1}B_{i-1}) \\
&\quad = \mathrm{H}(A_i|SS_1S_2CA_1B_1 \ldots A_{i-1}B_{i-1}) - \mathrm{H}(A_i|R_1SS_1S_2CA_1B_1 \ldots A_{i-1}B_{i-1}) \\
&\quad \leq \mathrm{H}(A_i|SS_1S_2CA_1B_1 \ldots A_{i-1}B_{i-1}) \\
&\quad \leq \mathrm{H}(A_i|SS_2CA_1B_1 \ldots A_{i-1}B_{i-1}) \ .
\end{aligned}$$

On the other hand, using Claim 7.11 we can write

$$\begin{aligned}
&\mathrm{I}(A_i; R_1 | S S_1 S_2 C A_1 B_1 \ldots A_{i-1} B_{i-1}) \\
&\quad = \mathrm{H}(R_1 | S S_1 S_2 C A_1 B_1 \ldots A_{i-1} B_{i-1}) - \mathrm{H}(R_1 | A_i S S_1 S_2 C A_1 B_1 \ldots A_{i-1} B_{i-1}) \\
&\quad = \mathrm{H}(R_1 | S_1 C A_1 B_1 \ldots A_{i-1} B_{i-1}) - \mathrm{H}(R_1 | A_i S_1 C A_1 B_1 \ldots A_{i-1} B_{i-1}) \\
&\quad = \mathrm{H}(A_i | S_1 C A_1 B_1 \ldots A_{i-1} B_{i-1}) - \mathrm{H}(A_i | R_1 S_1 C A_1 B_1 \ldots A_{i-1} B_{i-1}) \; .
\end{aligned}$$

Thus,

$$\begin{aligned}
&\mathrm{H}(A_i | S S_2 C A_1 B_1 \ldots A_{i-1} B_{i-1}) \\
&\quad \geq \mathrm{H}(A_i | S_1 C A_1 B_1 \ldots A_{i-1} B_{i-1}) - \mathrm{H}(A_i | R_1 S_1 C A_1 B_1 \ldots A_{i-1} B_{i-1}) \qquad (7.1)
\end{aligned}$$

Since A_i is completely determined by the random input R_1, the piece S_1 and the previous messages $C, A_1, B_1, \ldots, A_{i-1}, B_{i-1}$, it holds that $\mathrm{H}(A_i | R_1 S_1 C A_1 B_1 \ldots A_{i-1} B_{i-1}) = 0$. Combining this inequality and Inequality (7.1), we get

$$\mathrm{H}(A_i | S S_2 C A_1 B_1 \ldots A_{i-1} B_{i-1}) \geq \mathrm{H}(A_i | S_1 C A_1 B_1 \ldots A_{i-1} B_{i-1}) \; .$$

□

We now prove the analogue of Claim 7.8, without the restriction to deterministic reconstruction.

Claim 7.13: $I(C_{1,2}; S_1 | C) \geq H(S)$.

Proof: Expressing $\mathrm{I}(C_{1,2}; S_1 | C)$ as a sum and using the fact that mutual information is non-negative, we get

$$\begin{aligned}
\mathrm{I}(C_{1,2}; S_1 | C) &= \sum_{i=1}^{k} \mathrm{I}(A_i; S_1 | C A_1 B_1 \ldots A_{i-1} B_{i-1}) + \sum_{i=1}^{k} \mathrm{I}(B_i; S_1 | C A_1 B_1 \ldots A_{i-1} B_{i-1} A_i) \\
&\geq \sum_{i=1}^{k} \mathrm{I}(A_i; S_1 | C A_1 B_1 \ldots A_{i-1} B_{i-1}) \\
&= \sum_{i=1}^{k} \mathrm{H}(A_i | C A_1 B_1 \ldots A_{i-1} B_{i-1}) - \mathrm{H}(A_i | S_1 C A_1 B_1 \ldots A_{i-1} B_{i-1}) \\
&\geq \sum_{i=1}^{k} \mathrm{H}(A_i | S_2 C A_1 B_1 \ldots A_{i-1} B_{i-1}) - \mathrm{H}(A_i | S_1 C A_1 B_1 \ldots A_{i-1} B_{i-1}) \; .
\end{aligned}$$

This inequality together with Claim 7.12 imply

$$\mathrm{I}(C_{1,2}; S_1 | C) \geq \sum_{i=1}^{k} \mathrm{H}(A_i | S_2 C A_1 B_1 \ldots A_{i-1} B_{i-1}) - \mathrm{H}(A_i | S S_2 C A_1 B_1 \ldots A_{i-1} B_{i-1})$$

$$= \sum_{i=1}^{k} \mathrm{I}(A_i; S|S_2 C A_1 B_1 \ldots A_{i-1} B_{i-1})$$
$$= \sum_{i=1}^{k} \mathrm{H}(S|S_2 C A_1 B_1 \ldots A_{i-1} B_{i-1}) - \mathrm{H}(S|S_2 C A_1 B_1 \ldots A_{i-1} B_{i-1} A_i)\,.$$

Since the message B_i is determined by S_2 and previous messages,

$$\mathrm{H}(S|S_2 C A_1 B_1 \ldots A_{i-1} B_{i-1} A_i) = \mathrm{H}(S|S_2 C A_1 B_1 \ldots A_{i-1} B_{i-1} A_i B_i)\,.$$

Therefore the "internal" summands in the last sum cancel each other, and we are left with

$$\mathrm{I}(C_{1,2}; S_1|C) \geq \mathrm{H}(S|S_2 C) - \mathrm{H}(S|S_2 C C_{1,2})\ . \tag{7.2}$$

Recall that the communication C and the piece S_2 give no information (to P_2) about the secret, i.e. $\mathrm{H}(S|S_2 C) = \mathrm{H}(S)$. On the other hand, P_2 holding S_2 and knowing $C_{1,2}$, can reconstruct the secret, i.e. $\mathrm{H}(S|S_2 C C_{1,2}) = 0$. Therefore, $\mathrm{H}(S|S_2 C) - \mathrm{H}(S|S_2 C C_{1,2}) = \mathrm{H}(S)$. Together with Inequality (7.2), we get $\mathrm{I}(C_{1,2}; S_1|C) \geq \mathrm{H}(S)$, as claimed. □

Claim 7.13 implies that Theorem 7.10 holds also in the scenario in which the parties can toss coins during the reconstructions.

7.6 Conclusions

In this chapter we investigated the cost of performing the reconstruction over *public* communication channels. In Fig. 7.4 we summarize our results for the various schemes. We denote by ℓ the size of the secret, and the sizes of the pieces are multiples of ℓ (e.g. 10ℓ). We also give the minimum size of secrets for which this piece size applies.

t-out-of-n scheme	piece size	min. secret size
Naive	$2n\ell$	$\log n$
Non-interactive	$n\ell$	$\log n$
Unrestricted	$(1 + 4(n-1)/9)\ell$	$t \log n$
One-time	$10(\log(n/t) + 2)\ell$	$\max\{n \log n, 9n\}$
One-time, exactly t	10ℓ	$\max\{t \log n, 9t\}$

Figure 7.4: Summary of the complexity of our schemes

איור 7.4: סיכום של הסיבוכיות של הסכמות שלנו.

In Fig. 7.5 we give two examples of the sizes of pieces in the various schemes. In both examples we consider a system with 1024 parties.

	$n = 1024$; $t = 128$		$n = 1024$; $t = 8$	
	pieces' size	min. secret size	pieces' size	min. secret size
Naive scheme	2047ℓ	10 bits	2047ℓ	10 bits
Unrestricted scheme	33ℓ	160 bytes	513ℓ	10 bytes
One-time at least t scheme	40ℓ	1280 bytes	80ℓ	1280 bytes
One-time exactly t scheme	10ℓ	160 bytes	10ℓ	90 bytes

Figure 7.5: Numerical Examples.

איור 7.5: דוגמאות מספריות.

Chapter 8

Computing Functions of a Shared Secret

In this chapter we introduce and study threshold (t-out-of-n) secret sharing schemes with respect to a *family of functions* $\mathcal{F}$. Such schemes allow any set of at least t parties to privately reconstruct the value $f(s)$ of a (previously distributed) secret s (for any $f \in \mathcal{F}$). Smaller sets of players "know nothing" about the secret. The goal is to make the pieces as short as possible.

The rest of this chapter is organized as follows: In Section 8.1 we provide the definitions of threshold schemes with respect to a family of functions. Section 8.2 contains schemes for the family of linear functions. Section 8.3 uses these schemes to construct schemes for other families. Section 8.4 contains non-interactive schemes for the family of bit functions. Section 8.5 contains the characterization of ideal schemes in various models.

8.1 Definitions

We define t-out-of-n secret-sharing schemes for a family of functions $\mathcal{F}$. This is an extension of the traditional secret sharing definition (Definition 3.6), which enables private reconstruction of functions of the secret. That is, any set G of cardinality at least t can reconstruct the value $f(s)$ of a (previously distributed) secret s (for any $f \in \mathcal{F}$), while any smaller set B "knows nothing". We distinguish between three types of schemes depending on the way that the value $f(s)$ is reconstructed:

1. *interactive secure private channels* schemes – where the parties in G engage in a protocol which computes the value $f(s)$ via private channels.

2. *non-interactive private channels* schemes – where during the reconstruction of $f(s)$ each party in G sends via a secure private channel a single message (depending only on his piece) to each of the other parties in G.

3. *non-interactive public channels* schemes – where each party in G sends a single message to each of the other parties in G on a public channel. Again each message depends only on the piece of the party.[1]

In all cases, the reconstruction is *private*. Namely, any set B of less than t parties, does not get any additional information about s. The notion of getting no information on s that is not implied by f is a generalization of Definition 3.4:

Definition 8.1 [No Information that is Not Implied by f]: *A coalition B* has no information that is not implied by a function f *on a random variable X if for every two possible values x_1, x_2 of X such that $f(x_1) = f(x_2)$, and every value of $VIEW_B$:*

$$\Pr[\, VIEW_B \,|\, X = x_1 \,] \;=\; \Pr[\, VIEW_B \,|\, X = x_2 \,],$$

where the probability is taken over the random inputs of the dealer, and the random inputs of the parties outside the coalition.

We first consider one-time schemes with respect to a family of functions. In these scheme, in addition to the usual security requirement, it is required that, after the reconstruction of $f(s)$ by a set G (of size at least t), any coalition B of size less than t, which has heard the communication, does not have any information (from its view of the system) on s that is not implied by $f(s)$. Recall that the term "communication heard by the coalition" depends on the communication model: in the private channels model it only contains the messages sent to parties in $G \cap B$ and in the public channel model it contains all the messages exchanged between the parties in G. In the private channel model a disjoint coalition does not gain any information on the secret. However, in the public channel model a disjoint coalition is allowed to gain information on $f(s)$ although not participating in the reconstruction.

Definition 8.2 [One Time Secret Sharing Schemes for Families]: *A t-out-of-n* secret sharing scheme with respect to a family of functions $\mathcal{F}$ *is a secret sharing scheme* Π *(according to Definition 3.6), that in addition satisfies the following two conditions:*

Function reconstruction requirement *For any set G of size at least t, and any function $f \in \mathcal{F}$ the parties in G can evaluate the value $f(s)$ (G and f are the common input for the reconstruction). The scheme is an* interactive private channel scheme *if this is done using an interactive protocol via private channels. The scheme is a* non-interactive private channel scheme *if the reconstruction is done using a non-interactive protocol via private channels. The scheme is a* non-interactive public channel scheme *if the reconstruction is done using a non-interactive protocol via public channels.*

One-time function security requirement *For any set of size less than t, the reconstruction is* secure. *That is, for any $f \in \mathcal{F}$, for any set G ($|G| \geq t$) that reconstructed $f(s)$, and any coalition B of size $|B| \leq t-1$, the members of the coalition B do not have*

[1] We do not consider interactive public channels schemes in this work.

any information that is not implied by f on the secret s from $VIEW_B$, where the view of B contains their pieces, their local random inputs, and the communication heard by the parties in B during the reconstruction of $f(s)$.

Obviously, any public scheme can be transformed into a scheme in which each message is sent on private channels. On the other hand, if we take a non-interactive private channels scheme and send the messages on public channels then the security of the reconstruction is not guaranteed.

We next define unrestricted schemes in which an unrestricted number of functions can be securely reconstructed by possibly different sets.

Definition 8.3 [Unrestricted Secret Sharing Schemes for Families]: *An* unrestricted *t-out-of-n* secret sharing scheme with respect to a family of functions $\mathcal{F}$ *is a secret sharing scheme with respect to a family $\mathcal{F}$ with a stronger security requirement:*

Unrestricted function security requirement *Let $f_1, f_2, \ldots, f_d$ be any sequence of functions from $\mathcal{F}$ (a function can appear more than once in the sequence), and $G_1, G_2, \ldots, G_d \subseteq \{P_1, \ldots, P_n\}$ be sequence of sets of cardinality at least t (a set can appear more than once in the sequence). Assume that for every i the set G_i reconstructed $f_i(s)$. Then, any coalition B of cardinality at most $t-1$ should not gain any additional information on the secret in the following manner:*

- *In the private channel model the coalition B hears communications during the reconstruction of f_i only if $B \cap G_{i_1} \neq \emptyset$. That is, let $G_{i_1}, G_{i_2}, \ldots, G_{i_{d'}}$ be the sets that intersect with G. The view of the members of the coalition B is their pieces and the communication received by the parties in $B \cap G_{i_1}, \ldots, B \cap G_{i_{d'}}$ during the reconstruction of $f_{i_1}(s), f_{i_2}(s), \ldots, f_{i_{d'}}(s)$ respectively. We require that the coalition does have any information about the secret that is not implied by the function $f_{i_1} \circ f_{i_2} \circ \ldots, f_{i_{d'}}(s)$ (the concatenation of the d' functions).*

- *In the public channel model the coalition B hears all the communications during the reconstruction of all of the functions. That is, the view of the members of the coalition B is their pieces and all the communication that takes place. We require that the coalition does not have any information about the secret that is not implied by all the function $f_1 \circ f_2 \circ \ldots, f_d(s)$ (the concatenation of the d functions).*

A function f' is *a renaming of f* if there exists a one-to-one function g such that $f'(x) = g(f(x))$. Note that a secret sharing scheme enables the reconstruction of a renaming of a function f if and only if the scheme enables the reconstruction of f. Therefore, we shall ignore renamings of functions.

8.1.1 Families of Functions – Basic Examples

In this section we define certain families of functions over $\mathrm{GF}(2^\ell)$ which are of interest. Recall the definition of the operations in $\mathrm{GF}(2^\ell)$. Each element $a \in \mathrm{GF}(2^\ell)$ is an ℓ-bit string $a_{\ell-1} \ldots a_1 a_0$ which is represented by the polynomial $a_{\ell-1}x^{\ell-1} + \ldots + a_1 x^1 + a_0$. Addition and multiplication are as for polynomials, using the structure of $\mathrm{GF}(2)$ for the coefficients and reducing products modulo a polynomial $p(x)$ of degree ℓ, which is irreducible over $\mathrm{GF}(2)$. A function $f : \mathrm{GF}(2^\ell) \to \mathrm{GF}(2^\ell)$ is *linear* over $\mathrm{GF}(2)$ if $f(x+y) = f(x) + f(y)$ for every $x, y \in \mathrm{GF}(2^\ell)$. It follows that f is linear if and only if f is the exclusive-or of a sub-set of the bits of its input. Clearly $f(ax) = af(x)$ for $a \in \mathrm{GF}(2)$. (Notice that we *do not* require that $f(ax) = af(x)$ for $a \in \mathrm{GF}(2^\ell)$.) We define the family $\mathcal{LIN}_\ell$ to be the family of all linear functions of $\mathrm{GF}(2^\ell)$. Let $e_i : \mathrm{GF}(2^\ell) \to \mathrm{GF}(2)$ be the function that returns the i-th bit of x. Notice that $e_i(x+y) = e_i(x) + e_i(y)$, hence e_i is linear. We call the family $\{e_1, \ldots, e_\ell\}$ the *bit functions* and denote it by $\mathcal{BIT}_\ell$. We also consider the family $\mathcal{ALL}_\ell$ of all possible functions of the secret. Without loss of generality we assume that the range of the functions in the family have the same size as the domain of the secrets.

8.2 Schemes for the Linear Functions

8.2.1 An Interactive Scheme

In this section we show that Shamir's scheme over $\mathrm{GF}(2^\ell)$ [101] (described in Example 4.4) is also a secret sharing scheme with respect to the family $\mathcal{LIN}_\ell$ – the family of linear functions.

Theorem 8.4 [Basic interactive scheme]: *For every $\ell \geq 1$, there exists an* unrestricted *t-out-of-n secret sharing scheme with respect to the family $\mathcal{LIN}_\ell$ in which the secrets are of length ℓ and the pieces are of length* $\max\{\ell, \lceil \log(n+1) \rceil\}$. *For $t = n$ this is a non-interactive public channels scheme; for $t = 2$ this is a non-interactive private channels scheme; for $3 \leq t \leq n-1$ it is an interactive private channels scheme.*

Proof: The dealer uses Shamir's t-out-of-n secret sharing scheme over $\mathrm{GF}(2^{\ell'})$, where $\ell' = \max\{\ell, \lceil \log(n+1) \rceil\}$, to distribute the pieces. We show how to reconstruct every linear function f securely. Consider a reconstructing set G of t parties that wishes to reconstruct $f(s)$. By the properties of Shamir's scheme, there exists a linear combination of the pieces held by the parties in G which equals to the secret. That is, there exist constants $\beta_1, \ldots, \beta_t \in \mathrm{GF}(2^{\ell'})$ such that for every secret $s \in \mathrm{GF}(2^\ell) \subseteq \mathrm{GF}(2^{\ell'})$ and any pieces $\{s_1, \ldots, s_t\}$ in $\mathrm{GF}(2^{\ell'})$ dealt to the parties in G it holds that $s = \sum_{j=1}^{t} \beta_j s_j$. Since f is linear we have $f(s) = f(\sum_{j=1}^{t} \beta_j s_j) = \sum_{j=1}^{t} f(\beta_j s_j)$. For a party $P_j \in G$ let $x_j \triangleq f(\beta_j s_j)$. Therefore, $f(s)$ is simply the sum of the x_j's. Computing the sum privately can be done using the interactive protocol of Benaloh [15] (for formal definition of privacy and description of Benaloh's protocol the reader is referred to Appendix B).

For the function security requirement, consider a coalition B of size smaller than t. Since each x_j is computed locally, and $f(s)$ is computed using the private protocol of [15], the parties in B gain no information about the x_j's (and hence the pieces) of other parties in G except for their sum which is $f(s)$. Hence, they gain no information about s other than what follows from the value $f(s)$. Furthermore, this scheme has the unrestricted security requirement. Since the reconstruction is done using a private protocol, the members of B cannot distinguish between two vectors of pieces with the same values $f_i(s)$. Hence, even if one knows some functions of the secret, then another reconstruction of a function cannot reveal extra information.

In the special cases where $t = 2$ or $t = n$, the parties can reconstruct the secret without interaction by simply sending the message x_j. If $t = 2$ these messages have to be sent via private channels, and in the case $t = n$ the messages can be sent via public channels. In these cases every coalition B of size $t-1$ which heard the communication on the private channels is contained in the reconstructing set G, so there exists j such that $G = B \cup \{P_j\}$. The parties in the coalition know $f(s)$ which equals $\sum_{P_i \in G} x_i$, and of course know $\sum_{P_i \in G \setminus \{P_j\}} x_i$. Hence, the coalition can reconstruct x_j from $f(s)$ and its pieces. Therefore, the coalition does not gain any extra information from the message x_j. Any subset B' of size smaller than $t-1$ has less information on the secret than B, hence such a set also gains no extra information about the secret. □

For $3 \leq t \leq n-1$, the parties reconstruct $f(s)$ using an interactive private channels scheme. It can be shown that, for this particular scheme, interactive reconstruction is essential. Otherwise, some coalitions B that intersect the reconstructing set G, but are not contained in G, can get some additional information about s if they hear the values x_j's.

8.2.2 A Non-interactive Public Channels Scheme

Next, we present a public channel t-out-of-n scheme with respect to the linear functions, in which the length of the pieces is $\ell \log n + \ell$.

Theorem 8.5 [Basic non-interactive public channels scheme]: *For every $\ell \geq 1$, there exists an unrestricted t-out-of-n non-interactive public channel secret sharing scheme with respect to the family $\mathcal{LIN}_\ell$ in which the secrets are of length ℓ and the pieces are of length $\ell \log n + \ell$.*

Proof: The scheme is as follows: The dealer shares each bit independently using Shamir's scheme. Since sharing each bit requires $\lceil \log(n+1) \rceil$ bits, the total length of the piece of each party is $\ell \lceil \log(n+1) \rceil \leq \ell \log n + \ell$. Clearly, every bit of the secret can be securely reconstructed. To reconstruct other linear functions of the secret, we use the homomorphic property of Shamir's scheme defined by Benaloh [15]. Consider the scenario in which the parties have pieces of two secrets. Now, every party sums his two pieces and the parties of G reconstruct a secret according to Shamir's scheme from the new pieces. The reconstructed

secret in this case is the sum of the two original secrets. Furthermore, every coalition of size at most $t-1$ seeing the new pieces gains no information on the secrets other than what is implied by the sum of the secrets. Similarly, the parties can securely reconstruct every linear combination of the bits by applying this combination to each of the pieces, and sending the new pieces on the public channel. As every linear function is a linear combination of the bits, the theorem follows. □

Remark 8.6: Observe that in the public channels scheme there is no need for the set G to be given as input for the reconstruction: each party P_i which is available just outputs an appropriate linear combination of its pieces, and its identity P_i. This scenario addresses reliability issues.

8.3 Schemes for Other Families of Functions

We now show how to use the basic scheme (for linear functions) to construct schemes for any family of functions. Given a family of functions, we shall construct a family of linear functions (of a bigger domain) which will enable the reconstruction of the original functions. However, the length of pieces in the new scheme may be much larger than the length of secrets. Observe that any Boolean function $f : \mathrm{GF}(2^\ell) \to \mathrm{GF}(2)$ can be represented as a binary vector over GF(2) of length 2^ℓ whose i-th coordinate is $f(i)$. Similarly, any function $f : \mathrm{GF}(2^\ell) \to \mathrm{GF}(2^{\ell'})$ can be represented as an array of ℓ' binary vectors in which the j-th vector corresponds to the j-th bit function of $f(x)$. The rank of a family of functions $\mathcal{F}$ is the smallest c for which there exist Boolean functions $f_1, \ldots, f_c$ such that for every function $f \in \mathcal{F}$ there exists a renaming of it, f', such that each of the ℓ' vectors representing f' is a linear combination of $f_1, \ldots, f_c$. The rank of a family does not change if we add renamings of functions. So, we can assume that $\mathcal{F}$ contains all renamings of its functions.

Theorem 8.7: *Let $\mathcal{F}$ be any family of functions.*

- *There exists an unrestricted t-out-of-n* interactive *public channels secret sharing scheme with respect to $\mathcal{F}$ with pieces of length* $\max\{\mathrm{rank}(\mathcal{F}), \log n + 1\}$.
- *There exists an unrestricted t-out-of-n* non-interactive *public channels secret sharing scheme with respect to $\mathcal{F}$ with pieces of length* $\mathrm{rank}(\mathcal{F})(\log n + 1)$.

Proof: Let $f_1, \ldots, f_c$ be a basis for the vector space spanned by the functions in $\mathcal{F}$. To share a secret s, the dealer generates a new secret $\mathrm{E}(s) = f_1(s) \circ f_2(s) \circ \ldots \circ f_c(s)$ of length c (where $\circ$ denotes concatenation of strings). The dealer now shares the secret $\mathrm{E}(s)$ using the basic scheme. For every $i = 1, 2, \ldots, c$ it holds that $f_i(s) = e_i(\mathrm{E}(s))$. Let f be a function in $\mathcal{F}$ that should be reconstructed, and f' its renaming such that the vectors representing f' are spanned by $f_1, \ldots, f_c$. Every bit of $f'(s)$ is a linear combination of $f_1, \ldots, f_c$, and therefore

of the bits of $\mathrm{E}(s)$ (i.e. of $e_i(\mathrm{E}(s))$). Since addition in $\mathrm{GF}(2^c)$ is bitwise, concatenation of linear functions is a linear function too. Thus, $f'(s)$ can be computed by evaluating a linear function of $\mathrm{E}(s)$ (hence, f can be computed as well). By Theorems 8.4 and 8.5, the claim follows. □

We demonstrate the above construction by considering the family $\mathcal{ALL}$ of all possible functions of the secret.

Corollary 8.8:

- *There exists an unrestricted t-out-of-n interactive private channels secret sharing scheme with respect to $\mathcal{ALL}_\ell$ with pieces of length* $\max\left\{2^\ell - 1, \log n + 1\right\}$.
- *There exists an unrestricted t-out-of-n non-interactive public channels secret sharing scheme with respect to $\mathcal{ALL}_\ell$ with pieces of length* $(2^\ell - 1)(\log n + 1)$.

Proof: By Theorem 8.7, we have to show that the rank of $\mathcal{ALL}_\ell$ is $2^\ell - 1$. Let $c = 2^\ell - 1$. Consider the c functions $f_1, \ldots, f_c$, where $f_i(x) = 1 \Leftrightarrow x = i$. To reconstruct a Boolean function f, we notice that the functions $f(x)$ and $f(x) + 1$ are renamings of each other, so we can assume that $f(0) = 0$. Hence $f(s) = \sum_{1 \leq i \leq c} f_i(s) f(i)$, and therefore $f_1, \ldots, f_c$ form a basis for $\mathcal{ALL}_\ell$. In this case the secret s is encoded as the vector $\mathrm{E}(s)$ of length c in which the s-th coordinate is 1 and all the other coordinates are zero (with the exception $\mathrm{E}(0) = 0$). □

Notice that the length of the secret is ℓ, while the length of the pieces is $(2^\ell - 1)$. However, there are $2^{2^\ell - 1}$ different Boolean functions with domain of cardinality 2^ℓ (such that every function is not a renaming of another function). Therefore, our scheme is significantly better than the naive scheme in which we share every function (up to renaming) separately. Also the length of $E'(s)$ for $\mathcal{ALL}_\ell$ must be at least $\log 2^{2^\ell - 1} = 2^\ell - 1$, so the representation for $E'(s)$ that we use is the best possible for the family $\mathcal{ALL}_\ell$ in this particular scheme. It remains an interesting open question whether there exists a better scheme for this family, or whether one can prove that this scheme is optimal.

So far, we considered only *threshold* secret sharing schemes. Our definition of secret sharing with respect to a family $\mathcal{F}$ of functions can be naturally generalized for an arbitrary access structure. To construct such schemes, observe that most known schemes (e.g. [16, 62, 68, 105]) are linear: the piece of each party is a vector of elements over some field, and every set in $\mathcal{A}$ reconstructs the secret using a linear combination of elements in their pieces. Thus, if we share every bit of the secret independently, then we can reconstruct every linear function of the secret without any interaction (the details are as in Theorem 8.5). This implies that, for every access structure, there exists a scheme with respect to the family $\mathcal{ALL}_\ell$ in which the length of the pieces is $O(2^\ell 2^n)$. However, if the access structure has a more efficient scheme for sharing a single bit then the length of the pieces can be shorter (but at least 2^ℓ).

8.4 Non-interactive Public Channels Scheme for the Bit Family

In this section we present a *one-time* non-interactive public channels scheme for the family of bit functions, whose pieces are of length $O(\ell)$ (compared to $O(\ell \log n)$ of the unrestricted non-interactive scheme from Theorem 8.5).

Theorem 8.9: *Let $\ell > \log n \log\log n$. There exists a one-time non-interactive public channels t-out-of-n secret sharing scheme with respect to the family $\mathcal{BIT}_\ell$ in which the length of the secrets is ℓ and the length of the pieces is 3ℓ.*

We first present a meta-scheme (Section 8.4.1). Then, we show a possible implementation of the meta-scheme that satisfies the conditions of the theorem.

8.4.1 Meta Scheme

Let c and h be parameters (to be fixed later), where h is a prime power such that $h > \log n$, and the secret s is of length $\ell = ch$. We view the secret as a binary matrix with c rows and h columns. We construct, in a way that is described below, a new binary matrix H with $3c$ rows and h columns. Then we share every row of the matrix using Shamir's t-out-of-n scheme [101]. Since the length of every row is $h > \log n$, then Shamir's scheme is ideal and every party gets $3c$ pieces of length h. When a set G of cardinality t wants to reconstruct the bit $s_{i,j}$ of the secret, the parties in G reconstruct a subset $T_{i,j}$ of rows (which depends only on i, j and not on G). We will guarantee that these rows do not give any additional information on the secret.

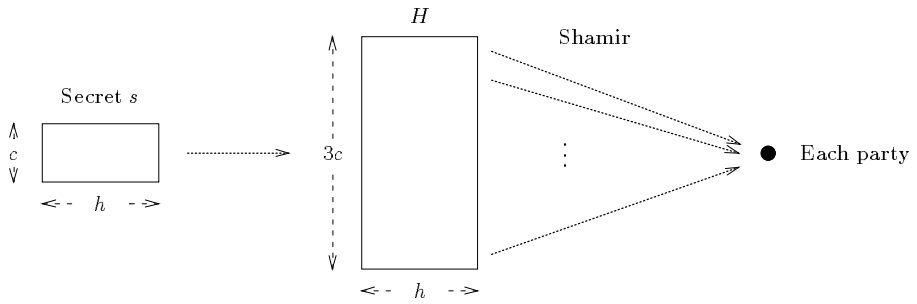

Figure 8.1: An illustration of the meta scheme for the family $\mathcal{BIT}_\ell$.
איור 8.1: תרשים של המטה-סכמה עבור המשפחה $\mathcal{BIT}_\ell$.

More specifically, for every $1 \leq i \leq c$ and $1 \leq j \leq h$, we fix a set $T_{i,j} \subseteq \{1, \ldots, 3c\}$ (independently of the secret). The j-th column of H is constructed independently, and it

depends only on the j-th column of the secret. It is chosen uniformly at random among the column vectors such that:

$$\forall_{1 \le i \le c} \sum_{a \in T_{i,j}} H_{a,j} = s_{i,j} \,. \tag{8.1}$$

To reconstruct $s_{i,j}$ it is enough to reconstruct the $T_{i,j}$-rows of H, and compute the sum (modulo 2) of the j-th bit of the reconstructed rows[2]. That is, the message of a party in a reconstructing set is the pieces corresponding to the $T_{i,j}$-rows of H. The existence of a matrix H satisfying Equation (8.1), and the one-time function security requirement depend on the choice of the sets $T_{i,j}$. On the other hand, independently of the choice of the sets $T_{i,j}$, any coalition of size $t-1$ prior to any reconstruction does not have any information on the rows of H (by the properties of Shamir's scheme), and hence does not have any information on the secret. That is, this scheme is a secret sharing scheme according to Definition 3.6 (this is one of the requirements for a scheme with respect to a family).

8.4.2 Implementing the Meta-Scheme

We show how to construct sets $T_{i,j}$ such that the security requirement after reconstruction will hold. Let $R_1, R_2, \ldots, R_h \subset \{c+1, \ldots, 3c\}$ be a collection of different sets of size c. In particular, this implies that no set is contained in another set. The number of sets of cardinality c which are contained in $\{c+1, \ldots, 3c\}$ is $\binom{2c}{c} \ge 2^{2c-o(c)} \ge 2^c$. Hence, we fix $c \triangleq \log h$ (that is $h \log h = \ell$). For $i = 1, \ldots, c$, let $T_{i,j} \triangleq R_j \cup \{i\}$. The following claims are useful for proving the security:

Claim 8.10: *For every secret s, the number of matrices H that satisfy Equation (8.1) for every j ($1 \le j \le h$) is at least 1, and is independent of s.*

Proof: Set $H_{i,j} = s_{i,j}$ for $1 \le i \le c$, and 0 otherwise. This matrix H satisfies Equation (8.1). To show that the number of matrices satisfying these equations is independent of s, notice that H is a solution to a non-homogeneous system of linear equations. Since the system has at least one solution, the number of such matrices is the number of solutions to the homogeneous linear system of equations (where every $s_{i,j} = 0$). □

To show that no coalition gains any additional information, we first consider the case in which the coalition knows only the $T_{i,j}$-rows of H. In this case the coalition can reconstruct $s_{i,j}$. We prove that the coalition does not gain any information on other bits of the secret (i.e., bits $s_{i',j'}$, for $i' \ne i$ or $j' \ne j$).

Claim 8.11: *Let s be a secret. Consider a setting of the $T_{i,j}$-rows of H such that $\sum_{a \in T_{i,j}} H_{a,j} = s_{i,j}$. The number of matrices H in which the $T_{i,j}$-rows are set and satisfy Equation (8.1) for every j ($1 \le j \le h$) is at least 1, and is independent of s.*

[2] Actually every party can sum the pieces of the $T_{i,j}$-rows of H, and then the parties reconstruct only the secret from these pieces. The j-th bit of the reconstructed secret equals to $s_{i,j}$.

Proof: We first prove that there exists a matrix H satisfying the requirements. Since the columns of H are constructed independently, it is enough to prove the existence of every column j'. Only the j'-th column of the $T_{i,j}$-rows of H, and the j'-th column of s influence the j'-th column of H, hence while considering the j'-th column we can ignore the rest of the columns.

We first consider the j-th column of H (i.e., $j' = j$). Since $T_{i',j} = R_j \cup \{i'\}$, we can assign $H_{i',j}$ a value as follows: for every i' such that $c+1 \leq i' \leq 3c$ and $i' \notin R_j$ set $H_{i',j} = 0$ (this is arbitrary); for every i' such that $1 \leq i' \leq c$, set $H_{i',j} = \sum_{a \in R_j} H_{a,j} + s_{i',j'}$. The case $j' \neq j$ is similar except that we have to be careful about $s_{i,j'}$. Since $R_{j'} \not\subseteq R_j$, there exists an element $p \in T_{i,j'} \setminus T_{i,j}$. we can assign $H_{i',j'}$ a value as follows: for every i' such that $c+1 \leq i' \leq 3c$, $i' \notin R_j$ and $i' \neq p$ set $H_{i',j'} = 0$; $H_{p,j'} = \sum_{a \in T_{i,j'} \setminus \{p\}} H_{a,j'} + s_{i,j'}$. For every i' ($1 \leq i' \leq c$ and $i' \neq i$) set $H_{i',j'} = \sum_{a \in R_{j'}} H_{a,j'} + s_{i',j'}$.

We have shown that there exists a matrix as required. By the same arguments as in the proof of Claim 8.10, the number of possible matrices given a secret s, is independent of s. □

We are now ready to prove the security requirement.

Lemma 8.12: *The above scheme is secure.*

Proof: Let G be any set that reconstructed a bit $s_{i,j}$ of the secret, and B be any coalition. The fact that this scheme is a secret sharing scheme according to Definition 3.6 is immediate (discussed at the end of Section 8.4.1). Suppose that B heard the messages sent by the parties in G. That is, the parties in B know the pieces of the parties of G corresponding to the $T_{i,j}$-rows of H. Therefore, the only information that they gained is the $T_{i,j}$-rows of H. That is, given the pieces of the coalition and all the messages that were sent, every matrix H that agrees with these rows is possible. We need to prove that given two secrets in which the (i,j)-th bit is the same, and a matrix H in which only the $T_{i,j}$-rows are fixed, then the probability that H was constructed from the secrets is equally likely. But by Claim 8.10 and Claim 8.11, this probability only depends on $s_{i,j}$. □

Note that the scheme is secure only for one reconstruction. If two bits are reconstructed then the rows that a coalition knows can contain the rows of a third bit. Therefore, the coalition learns this third bit as well.

8.5 Characterization of Families with Ideal Schemes

In this section we consider *ideal* secret sharing schemes with respect to a family $\mathcal{F}$. We say that a family $\mathcal{F}$ can *distinguish* between two secrets s, s' if there exists some function $f \in \mathcal{F}$ such that $f(s) \neq f(s')$. A scheme with respect to $\mathcal{F}$ is *ideal* if the cardinality of the domain of pieces equals the cardinality of the domain of distinguishable secrets[3]. Consider a family

[3]These are the shortest pieces possible by [69].

$\mathcal{F}$ of functions that contains all the bit functions $e_i(s)$. The following theorem gives several impossibility results with respect to such family $\mathcal{F}$ for which an ideal secret sharing exists. If interaction is allowed then every *Boolean* function in $\mathcal{F}$ is a linear functions (item (1) below). If the scheme is non-interactive the situation is even worse: for $3 \leq t \leq n-1$ such a scheme does not exist (item (3) below), for $t = 2$ and $t = n$ the family $\mathcal{F}$ contains only linear functions (item (2) below). Furthermore, for $t = 2$, reconstruction cannot be done on public channels (item (4) below). All these results hold even for one-time scheme (thus, for unrestricted schemes as well). This is formalized by the following theorem.

Theorem 8.13 [Characterization Theorem]:

1. *Let $2 \leq t \leq n$. Suppose that there is an ideal,* interactive, *t-out-of-n secret sharing scheme with respect to a family of functions $\mathcal{F}$ such that $\mathcal{BIT}_\ell \subseteq \mathcal{F}$. Let $f : \mathrm{GF}(2^\ell) \rightarrow \mathrm{GF}(2)$ be a* Boolean *function such that $f \in \mathcal{F}$. Then, f is a linear function.*

2. *Let $t = 2, n$. Suppose that there is an ideal,* non-interactive, *t-out-of-n secret sharing scheme with respect to a family of functions $\mathcal{F}$ such that $\mathcal{BIT}_\ell \subseteq \mathcal{F}$. Then, $\mathcal{F} \subseteq \mathcal{LIN}_\ell$.*

3. *Let $3 \leq t \leq n-1$. In every t-out-of-n ideal secret sharing with respect to $\mathcal{BIT}_\ell$, the reconstruction of every Boolean function requires interaction on private channel.*

4. *Let $t = 2$. In every 2-out-of-n ideal secret sharing with respect to $\mathcal{BIT}_\ell$, the reconstruction of every Boolean function requires private channels.*

We remark that by Theorem 8.4 all these results are "tight" (for sufficiently long secrets). In Section 8.5.1 we prove (1) and (2), and in Section 8.5.2 we prove (3) and (4).

8.5.1 Proofs of (1) and (2) of the Characterization Theorem

The proofs of (1) and (2) are similar, and are composed of two stages. We consider the following two-party scheme over $\mathrm{GF}(2^\ell)$, denoted XOR: Given a secret s, the dealer chooses uniformly at random $x \in \mathrm{GF}(2^\ell)$. The piece of the first party is x, and the piece of the second party is $y = s + x$. In the first stage we characterize the functions that can be reconstructed from the pieces in XOR. In the second stage, we show that if there exists a t-out-of-n, ideal secret sharing scheme for $\mathcal{F}$ then XOR is a secret sharing scheme with respect to $\mathcal{F}$. The combination of the two stages implies items (1) and (2).

Characterizing the Functions Reconstructible in XOR

We next characterize the functions that can be reconstructed in XOR *without interaction.* We prove that these functions are exactly the linear functions. Then, we characterize the Boolean functions that can be reconstructed in XOR *with interaction*, and show that these functions are exactly the Boolean linear functions.

Lemma 8.14: *Let f be any function that can be reconstructed without interaction in the scheme* XOR. *Then $f \in \mathcal{LIN}_\ell$.*

Proof: Assume, without loss of generality, that $f : \mathrm{GF}(2^\ell) \to \mathrm{GF}(2^\ell)$. (Otherwise consider the function of x defined as $\min\{y : f(y) = f(x)\}$, which is a renaming of f.) To prove that f is a renaming of a linear function we first prove (Claim 8.15) that $f(x+y)$ can be computed from $f(x)$ and $f(y)$. Then, we prove (Claim 8.16) that every function with this property is a renaming of a linear function.

Claim 8.15: *There exists a function g such that $f(x+y) = g(f(x), f(y))$.*

Proof: We use the following fact about the scheme XOR: Given that P_1 holds a piece $x \in \mathrm{GF}(2^\ell)$, the party P_2 can hold every piece $y \in \mathrm{GF}(2^\ell)$. Consider the message m sent by P_2 to P_1 while holding a piece y (if there are several possible messages choose, say, the lexicographically first). We claim that m is a renaming of $f(y)$. That is, $f(y_1) = f(y_2)$ if and only if the message m_1 sent by P_2 while holding y_1 is equal to the message m_2 sent by P_2 while holding y_2. If P_1 holds the piece 0 and P_2 holds the piece y then $s = y$ and obviously $f(s) = f(y)$. While holding two pieces y_1 and y_2 such that $f(y_1) \neq f(y_2)$, party P_2 does not know whether P_1 holds the piece 0, and must send different messages. On the other hand, if for two pieces y_1 and y_2 such that $f(y_1) = f(y_2)$ the party P_2 sends different messages then P_1, while holding the piece 0, receives additional information on s besides $f(s)$. Therefore, the message is a renaming of $f(y)$; without loss of generality, assume that this message is $f(y)$. Similarly, assume that the message sent by P_1 while holding a piece x is $f(x)$.

Hence, P_1 can reconstruct $f(s) = f(x+y)$ from x and $f(y)$. Moreover, for every two pieces x_1 and x_2 held by P_1 such that $f(x_1) = f(x_2)$ and every piece y, held by P_2, the party P_1 has to reconstruct the same value. This is true, since P_2 receives the same message in both cases and therefore reconstructs the same value for $f(s)$. Hence, P_1 can reconstruct $f(x+y)$ from $f(x)$ and $f(y)$, and this reconstruction function is the desired function g. □

Claim 8.16: *If there exists a function g such that $f(x+y) = g(f(x), f(y))$, for every $x, y \in \mathrm{GF}(2^\ell)$, then f is a renaming of a linear function.*

Proof: Let $X_0 \triangleq \{x : f(x) = 0\}$. Assume, without loss of generality, that $f(0) = 0$, i.e. $0 \in X_0$ (otherwise consider the function $f(x) + f(0)$). We first prove that

$$f(x_1) = f(x_2) \text{ if and only if } f(x_1 + x_2) = 0\,. \tag{8.2}$$

For the "only if" direction: $f(x_1 + x_2) = g(f(x_1), f(x_2)) = g(f(x_1), f(x_1)) = f(x_1 + x_1) = f(0) = 0$. Similarly, the "if" direction follows from the next simple equations:

$$f(x_1) = f(x_1 + 0) = g(f(x_1), f(0)) = g(f(x_1), f(x_1 + x_2)) = f(x_1 + x_1 + x_2) = f(x_2)\,.$$

Equation (8.2) implies that if $x_1, x_2 \in X_0$, then $x_1 + x_2 \in X_0$. That is, the set X_0 is a linear space over the field GF(2). Now, consider a linear transformation $f' : \mathrm{GF}(2^\ell) \to \mathrm{GF}(2^\ell)$ such that the null space of f' is X_0 (that is, $f'(x) = 0$ if and only if $x \in X_0$). Since X_0 is a linear space, such a linear transformation exists. We claim that f is a renaming of f'. We have to prove that $f(x) = f(y)$ if and only if $f'(x) = f'(y)$. By Equation (8.2), $f(x) = f(y)$ if and only if $x + y \in X_0$. By the definition, $x + y \in X_0$ if and only if $f'(x + y) = 0$. Since f' is linear, $f'(x) + f'(y) = f'(x + y)$. Hence, $f(x) = f(y)$ if and only if $f'(x) = f'(y)$. (Note that the fact that f can be reconstructed in XOR is not used in the proof of this claim.) □

To conclude, if $f(s)$ can be reconstructed from the pieces in XOR, then by Claim 8.15 there exists a function g such that $f(x + y) = g(f(x), f(y))$. By Claim 8.16, this implies that f is a renaming of a linear function. This completes the proof of Lemma 8.14. □

We now prove a similar lemma for interactive reconstruction. However, this lemma applies only to Boolean functions.

Lemma 8.17: *Let $f : \mathrm{GF}(2^\ell) \to \mathrm{GF}(2)$ be a* Boolean *function that can be reconstructed with interaction in the scheme* XOR. *Then f is a renaming of a linear function.*

Proof: Since XOR is an ideal scheme, any information that a party gets on the piece of the other party is translated to information on the secret. That is, the parties compute $f(s) = f(x+y)$ in a way that each party receives no information except of $f(x+y)$. In other words, they compute the Boolean function f privately. (For formal definition of privacy the reader is referred to Appendix B.) Bivariate Boolean private functions were characterized by Chor and Kushilevitz [40]:

Claim 8.18 [40]: *Let A_1, A_2 be nonempty sets and $f : A_1 \times A_2 \to \{0,1\}$ be an arbitrary Boolean function. Then f can be computed privately if and only if there exist Boolean functions $f_1 : A_1 \to \{0,1\}$, $f_2 : A_2 \to \{0,1\}$ such that for every $x_1 \in A_1$, $x_2 \in A_2$ it holds that $f(x_1, x_2) = f_1(x_1) + f_2(x_2)$.*

Hence, there exists Boolean functions f_1, f_2 such that $f(x+y) = f_1(x)+f_2(y)$. In particular, $f(x) = f(x + 0) = f_1(x) + f_2(0)$. Similarly, $f(y) = f_1(0) + f_2(y)$. Therefore, there exists $\sigma \in \mathrm{GF}(2)$ such that $f(x + y) = f(x) + f(y) + \sigma$. Thus, the function $f'(x) \triangleq f(x) + \sigma$ is a linear function which is a renaming of f. □

The exact family of functions that can be reconstructed in XOR with interaction is the two-argument functions such that $f(s) = f(x + y)$ can be computed privately from x and y (as characterized in [74]).

Reduction to XOR

We prove that if there exists an ideal t-out-of-n secret sharing scheme with respect to a family $\mathcal{F}$, then there exists a 2-out-of-2 ideal secret sharing scheme with respect to $\mathcal{F}$. Then, in turn, we prove that this implies that XOR is a secret sharing with respect to $\mathcal{F}$.

Claim 8.19: *Let $\mathcal{F}$ be a family of functions, and $2 \leq t \leq n$. If there exists an interactive (respectively non-interactive), ideal, t-out-of-n secret sharing scheme with respect to $\mathcal{F}$, then there exists an interactive (respectively non-interactive), ideal, 2-out-of-2 secret sharing scheme with respect to $\mathcal{F}$.*

Proof: Let $G = \{P_1, \ldots, P_t\}$ be a set of parties. We ignore the pieces distributed to parties not in G. Therefore, we have an ideal t-out-of-t secret sharing scheme with respect to $\mathcal{F}$. Let $\langle s_3, \ldots, s_t \rangle$ be any (fixed) vector of pieces that is dealt with positive probability in this scheme. The dealer now generates a vector of pieces (according to the scheme) that agree with $s_3, \ldots, s_t$ respectively, and gives the first two elements of the vector to P_1 and P_2 (respectively). Parties P_1 and P_2 know the pieces of the other parties (as they are fixed) and therefore, one of the parties, say P_1, can simulate the parties $P_3, \ldots, P_t$ in the protocol for reconstructing a function $f \in \mathcal{F}$. Hence, P_1 and P_2 can reconstruct $f(s)$ from their pieces and the messages they exchange. On the other hand, P_1 has no more information than the information known to the coalition $\{P_1, P_3, \ldots, P_t\}$ of cardinality $t-1$ in the t-out-of-t scheme. That is, P_1 does not gain additional information about s. Similar arguments hold for P_2. This implies that the scheme is a 2-out-of-2 secret sharing scheme with respect to $\mathcal{F}$. □

Lemma 8.20: *Let $\mathcal{F}$ be a family of functions such that $\mathcal{BIT}_\ell \subseteq \mathcal{F}$. If there exists an interactive (respectively non-interactive), ideal, t-out-of-n secret sharing scheme with respect to $\mathcal{F}$, then* XOR *is an interactive (respectively non-interactive) secret sharing scheme with respect to $\mathcal{F}$.*

Proof: By Claim 8.19, we can assume that there exists an ideal, 2-out-of-2 secret sharing scheme with respect to $\mathcal{F}$, denoted Π. We transform Π into XOR in a way that every function that can be reconstructed in Π can also be reconstructed in XOR.

Since the scheme is ideal, then given the piece x of P_1 there is a one-to-one and onto function from y, the piece of P_2, to s, the secret.[4] Therefore, each party must not gain any information on the piece of the other party beside the information he gets from the value $f(s)$. I.e. the computation of $f(s)$ has to be private. By Claim 8.18 [40], for every i there exist Boolean functions f_1^i, f_2^i such that $e_i(s) = f_1^i(x) + f_2^i(y)$. We define $\overline{\mathrm{m}}_1(x)$ as the vector of the values of these functions, that is $\overline{\mathrm{m}}_1(x) \triangleq f_1^1(x) \circ f_1^2(x) \circ \ldots \circ f_1^\ell(x)$, and $\overline{\mathrm{m}}_2(y) \triangleq f_2^1(y) \circ f_2^2(y) \circ \ldots \circ f_2^\ell(y)$. For every secret s and random input r of the dealer $s = \overline{\mathrm{m}}_1(\Pi_1(s,r)) + \overline{\mathrm{m}}_2(\Pi_2(s,r))$. Therefore, the scheme Π', defined as $\Pi'(s,r) = \langle \overline{\mathrm{m}}_1(\Pi_1(s,r)), \overline{\mathrm{m}}_2(\Pi_2(s,r)) \rangle$, is a scheme equivalent to XOR. We still have to show that in Π' the two parties can securely reconstruct every function in $\mathcal{F}$.

[4] The parties P_1 and P_2 can reconstruct the secret. Thus, for every piece x there exists a mapping h_x, that given a piece y of P_2 returns the reconstructed secret. Since P_1 does not have any information on the secret given his piece, the mapping is onto the domain of secrets. Since the scheme is ideal, then the domain and range of h_x have the same cardinality. That is, h_x is one to one.

We now prove that $\overline{\mathrm{m}}_1$ is a one to one transformation. Clearly, the two parties can reconstruct the secret in Π', while every single party knows nothing on the secret. Therefore, Π' is a 1-out-of-2 secret sharing scheme, and by [69], the cardinality of the domain of pieces is at least the cardinality of the domain of secrets. Therefore, the cardinality of the range of $\overline{\mathrm{m}}_1$ is at least 2^ℓ. The domain of $\overline{\mathrm{m}}_1$, which is the set of pieces of P_1, has cardinality 2^ℓ. Therefore, $\overline{\mathrm{m}}_1$ is a one to one transformation and hence it has an inverse. Therefore, P_1 can reconstruct the piece x from $\overline{\mathrm{m}}_1(x)$. Similarly, P_2 can reconstruct the piece y holding $\overline{\mathrm{m}}_2(y)$. This implies that parties P_1 and P_2, while holding $\overline{\mathrm{m}}_1(x)$ and $\overline{\mathrm{m}}_2(y)$ respectively, can securely reconstruct every function $f \in \mathcal{F}$. We have proved that every function $f \in \mathcal{F}$ can be reconstructed in Π' which is equivalent to XOR. Furthermore, if in the original scheme the reconstruction required no interaction, then also in XOR the reconstruction requires no interaction. □

Lemma 8.14, Lemma 8.17 and Lemma 8.20 complete the proof of items (1) and (2) of Theorem 8.13.

8.5.2 Proofs of (3) and (4) of the Characterization Theorem

Next we prove that in any ideal t-out-of-n secret sharing scheme ($2 \leq t \leq n-1$) with respect to $\mathcal{BIT}_\ell$ the reconstruction requires private channels. Furthermore, for $3 \leq t \leq n-1$, the reconstruction requires interaction. For the proof, assume towards a contradiction that there exists an ideal t-out-of-n scheme in which the reconstruction can be held on public channels. As in the proof of Claim 8.19, this implies that there exists an ideal 2-out-of-3 scheme with the same property (since $t < n$). Our first claim is that, without loss of generality, in the reconstruction of $f(s)$ every party sends a one bit message, such that the sum of the messages is the reconstructed value $f(s)$.

Claim 8.21: *Assume there exists an ideal 2-out-of-3 secret sharing scheme in which a Boolean function f can be securely reconstructed without interaction via public channels. Then, P_1 and P_2 can securely reconstruct $f(s)$ by sending one bit messages m_1 and m_2, such that $m_1 + m_2 = f(s)$.*

Proof: Since the parties P_1, P_2 can reconstruct the secret, there exists a function $h(\cdot,\cdot)$ that reconstructs the secret from the pieces s_1, s_2 of parties P_1, P_2 (respectively). Consider a possible message of P_1 when the set $\{P_1, P_2\}$ reconstructs the value $f(s)$ while P_2 holds the piece $s_2 = 0$. This message depends only on the piece of P_1 (P_1 might toss coins). Since the scheme is ideal, every piece must be possible for P_1 while $s_2 = 0$ (otherwise P_2 will know that some secret is not possible). When P_1 holds two pieces s_1 and s_1' such that $f(h(s_1, 0)) \neq f(h(s_1', 0))$, the possible messages have to be different or otherwise P_2 may reconstruct an incorrect value in one of the cases. On the other hand, for two pieces s_1 and s_1' such that $f(h(s_1, 0)) = f(h(s_1', 0))$ the possible messages of P_1 have to be the same or otherwise P_2 can distinguish between the secrets $s = h(s_1, 0)$ and $s' = h(s_1', 0)$ although

$f(s) = f(s')$. Thus, without loss of generality, the message m_1 of P_1 while holding the piece s_1 is the bit $f(h(s_1, 0))$. Similarly, the messages m_2 of P_2 while holding the piece s_2 is $f(h(0, s_2))$. Furthermore, the value of $f(s)$ is determine by these two messages (if P_2 would reconstruct two different values for $f(s)$ depending on his piece, P_1 will not know about it and will reconstruct the same value). There are two values for $f(s)$, and two values for each message. Since every party knows nothing about $f(s)$, every change of any message must change the value of $f(s)$. The only Boolean functions with two binary variables satisfying this requirement are addition modulo 2; i.e., $m_1 + m_2 + \sigma$ with either $\sigma = 0$ or $\sigma = 1$. So, without loss of generality, assume that P_1 sends the message $m_1 + \sigma$, and the lemma follows. □

Lemma 8.22: *Let f be any Boolean function, and consider an ideal 2-out-of-3 secret sharing scheme with respect to a family $\mathcal{F}$ which contains f. Then f cannot be reconstructed on public channels without interaction.*

Proof: Consider the case where the piece s_3 of P_3 is 0. In this case, the piece s_1 is a permutation of the secret. Hence, when we refer to the message that P_1 sends while the secret is s, we mean the messages that is sent for the piece that corresponds to the secret s (of course P_1 does not know that $s_3 = 0$ and does not know the secret). For every two secrets s, s' such that $f(s) = f(s')$, the messages of P_1, while P_1 and P_2 reconstruct the secret, should be the same (otherwise P_3 will differ between the two secrets). That is, for some value c, the value of $f(s)$ is 0 if and only if P_1 sends the message c, and $f(s) = 1$ if and only if P_1 sends the message $c + 1$. By Claim 8.21 $f(s) = m_1 + m_2$, so the message of P_2 has to be c for both values of $f(s)$. That is, while $s_3 = 0$ either $f(s)$ is constant and P_3 will know it prior to any reconstruction, or $f(s)$ is not constant and P_1 will not be able to reconstruct it from its piece and the message of P_2. Both cases lead to contradiction, hence $f(s)$ cannot be reconstructed without interaction on public channels. □

Lemma 8.23: *Assume there exists an ideal 3-out-of-4 secret sharing scheme in which f can be reconstructed via private channels without interaction. Then there exists an ideal 2-out-of-3 secret sharing scheme in which f can be reconstructed via public channels without interaction.*

Proof: By the same arguments as in Claim 8.21, we can assume that the messages sent by each party in the reconstruction to the other two parties are identical (that is, the message is $f(h(s_1, 0, 0))$ where h is a function that reconstructs the secret from the three pieces). Therefore, if we publish the piece of P_4 and send the messages on public channels, then every set of size 2 can reconstruct $f(s)$. Every party P_i will have the same information as the coalition $\{P_i, P_4\}$ had in the original scheme. Hence P_i does not gain extra information. That is, we got a 2-out-of-3 secret sharing scheme with respect to $\mathcal{F}$. □

Chapter 9

Conclusions and Open Problems

This thesis dealt with two fundamental cryptographic tools: secret sharing schemes and key distribution schemes. Secret sharing schemes enable only some predefined sets of parties to reconstruct a given secret. These schemes make it possible to store secret information in a network, such that only "good" (for example, large enough) subsets can reconstruct the information. Furthermore, by using these schemes we can allow only "good" subsets to perform actions in a system (e.g. sign a check). The scheme is called a t-out-of-n secret sharing scheme if any set of size t can reconstruct the secret, while every smaller set knows nothing about the secret. Key distribution schemes enable every subset of parties to generate a secret key (different subsets have different keys). For example, these keys can be used in private key cryptosystems or for authentication.

Both schemes are used in a multiuser system. In this thesis we considered (bad) parties which have unlimited power, i.e. we considered the information theoretic setting (in contrast to the setting in which parties are limited to probabilistic polynomial time computations). In the two cases we assumed that there exists an off-line dealer which distributes private pieces of information to the parties, when the system is initialized. The question we addressed is how long these pieces should be. This question is important since in some cases the size of the pieces is "big" (e.g. exponential in the number of parties in the system), and the schemes are not practical. The (space) efficiency of a scheme was defined to be the size of the pieces in the scheme.

We considered key distribution schemes of three types. The first type is non-communicating schemes in which each party reconstructs the keys from its piece without communicating with other parties. Blundo et. al. [29] presented a scheme in which the size of the pieces is $\binom{g+b-1}{g-1}$ times the size of the key, where g is the size of sets that can reconstruct a key and b is the size of bad coalitions. Furthermore, in [29] it was proved that this size is also a lower bound. Therefore, the space efficiency of these schemes is completely understood, and for big values of b and g these schemes are not practical. We have given a new proof of this lower bound. The second type is unrestricted communicating schemes in which the parties can communicate during the reconstruction. However, it is required

that any coalition that heard the communication during many reconstructions does not gain any information on a key of a disjoint set. We have proved that the space efficiency of unrestricted communicating schemes is the same as non-communicating schemes, and for big values of b and g they are not practical as well. The third type is one-time communicating schemes in which the security is guaranteed only if one set (whose identity is not known during the distribution stage) generates a key. We presented one-time schemes in which the size of the pieces is only $2(1 + (b-1)/g)$ times the size of the key. We complemented this result by proving that the size of the pieces in one-time key distribution schemes is at least $b/(g-1)$ times the size of the key. Therefore, the size of the pieces in our scheme is optimal up to a factor of two. Our schemes was generalized in [33], and for when $b \leq g-3$ they gave slightly more efficient schemes. Specificly, for $b = 1$ the efficiency of their scheme is $\approx 1 + 2/\sqrt{g}$ times the size of the keys (compared to 2 times the size of keys in our scheme).

We also considered t-out-of-n secret sharing scheme with *public reconstruction*. In these schemes we assumed that reconstruction of the secret by sets of at least t parties takes place over public channels. It was required that a disjoint coalition of less than t parties will not gain any information on the secret although it heard the communication exchanged in the system. A naive implementation of this task distributes $O(n)$ one times pads to each party. This results in pieces whose size is $O(n)$ times the secret size. We presented three implementations of such schemes that are substantially more efficient: (1) A scheme enabling multiple reconstructions of the secret by different subsets of parties, with factor $O(n/t)$ increase in the pieces' size. (2) A one-time scheme, enabling a single reconstruction of the secret, with $O(\log(n/t))$ increase in the pieces' size. (3) A one-time scheme, enabling a single reconstruction by a set of size *exactly* t, with factor $O(1)$ increase in the pieces' size. We proved that the first implementation is optimal (up to constant factors) by showing a tight $\Omega(n/t)$ lower bound for the increase in the pieces' size. The third implementation is also optimal (up to constant factors) by the lower bound of [69].

There are still some open problems concerning secret sharing with public reconstruction. It is an open question whether the second implementation is optimal, and if the $O(\log(n/t))$ increase in the pieces' size is essential. Another open question concerns non-interactive schemes, in which the messages sent by each party depend only on his piece (and not on messages received during the reconstruction). We presented a non-interactive, unrestricted t-out-of-n schemes, in which the size of the pieces is n times the size of the secret. However, the only lower bound we know is $\Omega(n/t)$ times the size of the secret, which applies to interactive schemes as well. The open question is to find better non-interactive schemes or improve the lower bound.

We considered t-out-of-n secret sharing schemes with respect to a *family of functions* $\mathcal{F}$. Such schemes allow any set of at least t parties to privately reconstruct the value $f(s)$ of a (previously distributed) secret s (for any $f \in \mathcal{F}$). Smaller sets of players "know nothing" about the secret. Such schemes contain as special cases multi-secret sharing schemes in which many secrets are shared simultaneously and threshold cryptology. We defined the notion of secret sharing with respect to a family of functions, and showed examples in which these

schemes are useful. We have showed an interactive scheme for the family of linear functions in which the size of the pieces is the same as the size of the secret. We have used this schemes to construct a scheme for every family of functions. For the most general family – $\mathcal{ALL}_\ell$ – of all functions, the pieces in our scheme were 2^ℓ bit long.

This work is only the first step in understanding the efficiency of secret sharing schemes with respect to a family of functions. If one can construct an efficient scheme for the family $\mathcal{ALL}$ then the question of the efficiency of these schemes will be resolved. So, the immediate open question is to construct an efficient scheme for the family $\mathcal{ALL}$, or prove that such scheme cannot exist (as we suspect). There are other families of functions for which it would be interesting to design efficient schemes (which might exist even if $\mathcal{ALL}$ does not have an efficient schemes). For example, an interesting family is the family of the sub-string functions – is a given string a sub-string of the secret (the family contains a function for every given string).

Finally, we considered generalized secret sharing schemes in which there is some monotone collection $\mathcal{A}$ of subsets of the parties, called the access structure. It is required that every set in $\mathcal{A}$ can reconstruct the secret, while every set not in $\mathcal{A}$ has no information on the secret. For most access structures the size of the pieces in the known schemes is exponential in the number of parties. Therefore, these schemes are not practical. Understanding if exponential pieces are essential or there are more efficient schemes is an important question. We conjectured that there exist access structures for which in every secret sharing scheme realizing the access structure the size of the pieces is exponential (Conjecture 1.1). However, no proof of the existence of such access structure is known to date even for a non-explicit access structure. Unlike other complexity measures, e.g. circuit complexity, we do not know that the number of efficient schemes is small compared to the number of monotone access structures (which is double exponential). So, proving Conjecture 1.1 even for a non-explicit access structure is an important open problem.

Since we were not able to prove this conjecture, we focused on linear secret sharing schemes. This class of schemes contains most known schemes to date. Lower bounds for linear schemes are a first step for proving lower bounds for general schemes. Furthermore, they explain the limitations of existing schemes. We presented a new technique for proving lower bounds for monotone span programs, and used it to prove an $\Omega(n^{2.5})$ lower bound for an explicit access structure with n parties. A recent result [4] uses our technique (Theorem 5.6), and proves a super-polynomial lower bounds for the size of linear secret sharing scheme for an access structure that is related to a problem in extremal set theory. It remains open whether our method could yield exponential lower bounds. Thus, it is still an open problem to prove exponential lower bounds for linear secret sharing schemes realizing an *explicit* access structure. These can be done either by constructing an exponential size critical family and applying Theorem 5.6, or by developing a new technique.

A natural question is whether one can prove super-polynomial lower bounds for other classes of secret sharing schemes. Specificly, say that a secret sharing scheme is communication-ideal if for every $G \in \mathcal{A}$ each party of G can compute from his piece a

message whose size is the same as the size of the secret, and the secret can be reconstructed from these messages (without knowing any other information on the pieces). For different sets the same party is allowed to compute different messages. Although the pieces might be big, the reconstruction algorithm might only require each party to send short messages. For example, in a linear scheme over GF(q), it is enough that each party in the reconstructing set will send one element from GF(q) and the secret would be the sum of these elements. Hence, in linear schemes the size of each message can be the same as the size of the secret although the pieces can be exponential. The open question we suggest is how to generalize the proofs of the lower bounds for linear schemes such that they would apply for ideal-communication schemes as well. This question is interesting since the proof of the lower bounds for linear schemes relies on linear algebra arguments, and proving the lower bound for communication-ideal schemes would, presumably, contain different arguments. We remark that by the results of [8] every communication-ideal scheme with domain of secrets of size two or three is linear and the lower bounds apply to them as well.

We already claimed that based on some complexity assumptions we can prove that there exists an explicit monotone function in P that has no efficient linear scheme. Another open problem is to prove this property without any assumptions. That is, prove super-polynomial lower bound for linear schemes for an explicit function in P. We can make this question even harder: we know that every function that has an efficient linear scheme is in NC. Can we prove that the converse does not hold? That is, prove super-polynomial lower bound for linear schemes for an explicit monotone function in NC. An interesting candidate is the perfect matching function (or even perfect matching in a bipartite graph). This function has NC-circuits [84]. However, it does not have monotone circuits [94, 110].

It is known that efficient linear schemes exist for every function that has small undirected branching program [17, 68] (see Example 4.5). It is an open question if such schemes exist for directed branching programs as well. This question can be described as follows. Consider an access structure whose parties are the edges of a *directed* complete graph with two special vertices s and t. A set of parties (edges) is an authorized set if the sub-graph that contain these edges has a directed path from s to t. Again, this problem is known to be in NC^2 so we do not have evidence that this access structure does not have efficient linear schemes. Furthermore, Wigderson [112] proved that this function has small span programs. However, this span program is not monotone and its construction uses this non-monotonicity in a strong way. We do not know of an efficient linear scheme for the directed connectivity problem, and it is not clear if it exists.

The lower bounds we proved for linear secret sharing schemes raise the question if there are efficient non-linear schemes. It is not hard to construct non-linear schemes, e.g. [109, 35]. The open question is to find a non linear scheme realizing an access structure which does not have an efficient linear scheme. For example, find an efficient scheme for a function that (under reasonable assumptions) is not in NC.

Computational Secret Sharing In the secret sharing schemes that we considered the security is guaranteed even if the "bad" parties have unlimited power. These schemes are called perfect schemes. It is reasonable to assume that the parties are computationally limited, i.e. they run in (probabilistic) polynomial time. (This is the usual assumption in modern cryptography.) It is possible that there are access structures that have efficient computational secret sharing schemes, but do not have efficient perfect secret sharing schemes. Some results is this direction were shown by Yao [114]. Yao showed efficient computational schemes for access structures whose characteristic function can by computed by polynomial monotone *circuits* (as opposed to polynomial monotone formulae in the perfect schemes of [16]). Specificly, there are access structures that have efficient computational secret sharing schemes but are not known to have efficient perfect schemes. The open question is which access structures have efficient computational schemes, and what is the exact cryptographic assumption that are needed for them.

Bibliography

[1] M. Abadi, J. Feigenbaum, and J. Kilian. On hiding information from an oracle. *Journal of Computer and System Sciences*, 39:21–50, 1989.

[2] R. Ahlswede and I. Csiszár. Common randomness in information theory and cryptography – part i: Secret sharing. *IEEE Trans. on Information Theory*, 39(4):1121–1132, 1993.

[3] N. Alon and R. B. Boppana. The monotone circuit complexity of Boolean functions. *Combinatorica*, 7(1):1–22, 1987.

[4] L. Babai, A. Gál, J. Kollár, L. Rónyai, T. Szabó, and A. Wigderson. Extremal bipartite graphs and superpolynomial lower bounds for monotone span programs. In *28th STOC*, 1996.

[5] D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. In C. Choffrut and T. Lengauer, editors, *STACS 90, 7th Annual Symposium on Theoretical Aspects of Computer Science, Proceedings*, volume 415 of *Lecture Notes in Computer Science*, pages 37–48. Springer-Verlag, 1990.

[6] A. Beimel. Ideal secret sharing schemes. Master's thesis, Technion - Israel Institute of Technology, Haifa, 1992. (In Hebrew, Abstract in English).

[7] A. Beimel, M. Burmester, Y. Desmedt, and E. Kushilevitz. Computing functions of a shared secret. Submitted for publication, 1995.

[8] A. Beimel and B. Chor. Universally ideal secret sharing schemes. *IEEE Trans. on Information Theory*, 40(3):786–794, 1994.

[9] A. Beimel and B. Chor. Secret sharing with public reconstruction. In D. Coppersmith, editor, *Advances in Cryptology - CRYPTO '95*, volume 963 of *Lecture Notes in Computer Science*, pages 353–366. Springer-Verlag, 1995.

[10] A. Beimel and B. Chor. Communication in key distribution schemes. *IEEE Trans. on Information Theory*, 42(1):19–28, 1996. An extended abstract in "Crypto '93", LNCS 773, pp. 444–455.

[11] A. Beimel, A. Gál, and M. Paterson. Lower bounds for monotone span programs. In *Proc. 36nd IEEE Symp. on Foundations of Computer Science*, pages 674–681, 1995.

[12] M. Bellare and P. Rogaway. Entity authentication and key distribution. In D. R. Stinson, editor, *Advances in Cryptology - CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 232–249. Springer-Verlag, 1994.

[13] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for noncryptographic fault-tolerant distributed computations. In *Proceeding 20th Annual Symposium on the Theory of Computing*, pages 1–10. ACM, 1988.

[14] M. Ben-Or and T. Rabin. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceeding 21th Annual Symposium on the Theory of Computing*, pages 73–85. ACM, 1989.

[15] J. Benaloh. Secret sharing homomorphisms: Keeping shares of a secret secret. In A. M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 251–260. Springer-Verlag, 1987.

[16] J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In S. Goldwasser, editor, *Advances in Cryptology - CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 27–35. Springer-Verlag, 1990.

[17] J. Benaloh and S. Rudich. Private communication, 1989.

[18] S. J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Inform. Process. Lett.*, 18:147–150, 1984.

[19] M. Bertilsson and I. Ingemarsson. A construction of practical secret sharing schemes using linear block codes. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology - AUSCRYPT '92*, volume 718 of *Lecture Notes in Computer Science*, pages 67–79. Springer-Verlag, 1993.

[20] Bird, Gopal, A. Herzberg, Janson, S. Kutten, R. Molva, and M. Yung. The kryptoknight family of light-weight protocols for authentication and key distribution. *IEEE Transactions on Networking*, 3, 1995.

[21] B. Blakley, G. R. Blakley, A. H. Chan, and J. Massey. Threshold schemes with disenrollment. In E. F. Brickell, editor, *Advances in Cryptology - CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 540–548. Springer-Verlag, 1993.

[22] G. R. Blakley. Safeguarding cryptographic keys. In *Proc. AFIPS 1979 NCC, vol. 48*, pages 313–317, June 1979.

[23] G. R. Blakley and C. Meadows. The security of ramp schemes. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology - CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, pages 242–268. Springer-Verlag, 1985.

[24] R. Blom. An optimal class of symmetric key generation systems. In T. Beth, N. Cot, and I. Ingemarsson, editors, *Advances in Cryptology – Eurocrypt '84*, volume 209 of *Lecture Notes in Computer Science*, pages 335–338. Springer-Verlag, 1985.

[25] C. Blundo, A. Cresti, A. De Santis, and U. Vaccaro. Fully dynamic secret sharing schemes. In D. R. Stinson, editor, *Advances in Cryptology - CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 110–125. Springer-Verlag, 1994.

[26] C. Blundo, A. De Santis, G. Di Crescenzo, A. Giorgio Gaggia, and U. Vaccaro. Multi-secret sharing schemes. In Y. Desmedt, editor, *Advances in Cryptology - CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 150–163. Springer-Verlag, 1994.

[27] C. Blundo, A. De Santis, L. Gargano, and U. Vaccaro. On the information rate of secret sharing schemes. *Theoretical Computer Science*, 154(2):283–306, 1996.

[28] C. Blundo, A. De Santis, A. Giorgio Gaggia, and U. Vaccaro. New bounds on the information rate of secret sharing schemes. *IEEE Trans. on Information Theory*, 41(2):549–553, 1995.

[29] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-secure key distribution for dynamic conferences. In E. F. Brickell, editor, *Advances in Cryptology - CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 471–486. Springer-Verlag, 1993.

[30] C. Blundo, A. De Santis, D. R. Stinson, and U. Vaccaro. Graph decomposition and secret sharing schemes. *Journal of Cryptology*, 8(1):39–64, 1995.

[31] C. Blundo, A. De Santis, and U. Vaccaro. Efficient sharing of many secrets. In P. Enjalbert, A. Finkel, and K. W. Wagner, editors, *STACS '93*, volume 665 of *Lecture Notes in Computer Science*, pages 692–703. Springer-Verlag, 1993.

[32] C. Blundo, A. De Santis, and U. Vaccaro. Randomness in distribution protocols. In S. Abiteboul and E. Shamir, editors, *21st Annual International Colloquium on Automata, Languages and Programming*, volume 820 of *Lecture Notes in Computer Science*, pages 568–579. Springer-Verlag, 1994.

[33] C. Blundo, L. A. Frota Mattos, and D. R. Stinson. Some generalizations of the Beimel-Chor scheme, 1995. Unpublished manuscript, to be found in: http://bibd.unl.edu/~stinson/#bibs.

[34] E. F. Brickell. Some ideal secret sharing schemes. *Journal of Combin. Math. and Combin. Comput.*, 6:105–113, 1989.

[35] E. F. Brickell and D. M. Davenport. On the classification of ideal secret sharing schemes. *Journal of Cryptology*, 4(73):123–134, 1991.

[36] E. F. Brickell and D. R. Stinson. Some improved bounds on the information rate of perfect secret sharing schemes. *Journal of Cryptology*, 5(3):153–166, 1992.

[37] G. Buntrock, C. Damm, H. Hertrampf, and C. Meinel. Structure and importance of the logspace-mod class. Math. Systems Theory, 25:223–237, 1992.

[38] R. M. Capocelli, A. De Santis, L. Gargano, and U. Vaccaro. On the size of shares for secret sharing schemes. *Journal of Cryptology*, 6(3):157–168, 1993.

[39] D. Chaum, C. Crepau, and I. Damgard. Multiparty unconditionally secure protocols. In *Proceeding 20th Annual Symposium on the Theory of Computing*, pages 11–19. ACM, 1988.

[40] B. Chor and E. Kushilevitz. A zero-one law for Boolean privacy. *SIAM Journal of Disc. Math.*, 4(1):36–47, 1991.

[41] B. Chor and E. Kushilevitz. Secret sharing over infinite domains. *Journal of Cryptology*, 6(2):87–96, 1993.

[42] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, 1991.

[43] L. Csirmaz. The dealer's random bits in perfect secret sharing schemes, 1994. Preprint.

[44] L. Csirmaz. The size of a share must be large. In A. De Santis, editor, *Advances in Cryptology – Eurocrypt '94*, volume 950 of *Lecture Notes in Computer Science*, pages 13–22. Springer-Verlag, 1995.

[45] I. Csiszar and J. Körner. *Information Theory. Coding Theorems for Discrete Memoryless Systems.* Academic press, 1986.

[46] A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung. How to share a function securely. In *Proceedings of the twenty-sixth annual ACM Symp. Theory of Computing (STOC)*, pages 522–533, 1994.

[47] A. De Santis, G. Di Crescenzo, and G. Persiano. Secret sharing and perfect zero knowledge. In D. R. Stinson, editor, *Advances in Cryptology – CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 73–84. Springer-Verlag, 1994.

[48] Y. Desmedt. Threshold cryptosystems. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology - AUSCRYPT '92*, volume 718 of *Lecture Notes in Computer Science*, pages 5–14. Springer-Verlag, 1993.

[49] Y. Desmedt and Y. Frankel. Shared generation of authenticators and signatures. In J. Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 457–469. Springer-Verlag, 1992.

[50] Y. Desmedt and Y. Frankel. Homomorphic zero-knowledge threshold schemes over any finite abelian group. *SIAM Journal of Disc. Math.*, 7(4):667–679, 1994.

[51] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. on Information Theory*, 22(6):644–654, 1976.

[52] M. van Dijk. A linear construction of perfect secret sharing schemes. In A. De Santis, editor, *Advances in Cryptology – Eurocrypt '94*, volume 950 of *Lecture Notes in Computer Science*, pages 23–34. Springer-Verlag, 1995.

[53] M. van Dijk. On the information rate of perfect secret sharing schemes. *Designs, Codes and Cryptography*, 6:143–169, 1995.

[54] A. Fiat and M. Naor. Broadcast encryption. In D.R. Stinson, editor, *Advances in Cryptology - CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer-Verlag, 1994.

[55] M. J. Fischer, M. S. Paterson, and C. Rackoff. Secure bit exchange using a random deal of cards. In *Distributed Computing and Cryptography*, pages 173–181. AMS, 1991.

[56] M. J. Fischer and R. N. Wright. Multiparty secret key exchange using a random deal of cards. In J. Feigenbaum, editor, *Advances in Cryptology – CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 141–155. Springer-Verlag, 1992.

[57] M. J. Fischer and R. N. Wright. An efficient protocol for unconditionally secure secret key exchange. In *4th Annual Symposium on Discrete Algorithms*, pages 475–483, 1993.

[58] M. J. Fischer and R. N. Wright. Bounds on secret key exchange using a random deal of cards. *Journal of Cryptology*, 9(2):71–99, 1996.

[59] M. Franklin and M. Yung. Communication complexity of secure computation. In *Proceeding 24th Annual Symposium on the Theory of Computing*, pages 699–710. ACM, 1992.

[60] R. G. Gallanger. *Information Theory and Reliable Communications.* John Wiley & Sons, New York, 1968.

[61] L. Gong and D. J. Wheeler. A matrix key-distribution scheme. *Journal of Cryptology*, 2(1):51–59, 1990.

[62] M. Ito, A. Saito, and T. Nishizeki. Secret sharing schemes realizing general access structure. In *Proc. IEEE Global Telecommunication Conf., Globecom 87*, pages 99–102, 1987.

[63] W. Jackson and K. M. Martin. On ideal secret sharing schemes. Unpublished manuscript, 1992.

[64] W. Jackson and K. M. Martin. Geometric secret sharing schemes and their duals. In *Designs, Codes and Cryptography*, volume 4, pages 83–95, 1994.

[65] W. Jackson, K. M. Martin, and C. M. O'Keefe. Multisecret threshold schemes. In D. R. Stinson, editor, *Advances in Cryptology - CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 126–135. Springer-Verlag, 1994.

[66] W. Jackson, K. M. Martin, and C. M. O'Keefe. On sharing many secrets. In J. Pieprzyk and R. Safavi-Naini, editors, *ASIACRYPT '94*, volume 917 of *Lecture Notes in Computer Science*, pages 42–54. Springer-Verlag, 1995.

[67] M. Karchmer. On proving lower bounds for circuit size. In *Proceeding of the 8th Annual Structure in Complexity Theory*, pages 112–118, 1993.

[68] M. Karchmer and A. Wigderson. On span programs. In *Proceeding of the 8th Annual Structure in Complexity Theory*, pages 102–111, 1993.

[69] E. D. Karnin, J. W. Greene, and M. E. Hellman. On secret sharing systems. *IEEE Trans. on Information Theory*, 29(1):35–41, 1983.

[70] T. Kővári, V. T. Sós, and P. Turán. On a problem of K. Zarankiewicz. *Colloq. Math.*, 3:50–57, 1954.

[71] J. Kilian and N. Nisan. Private communication, 1990.

[72] S. C. Kothari. Generalized linear threshold scheme. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology - CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, pages 231–241. Springer-Verlag, 1985.

[73] H. Krawczyk. Secret sharing made short. In D. R. Stinson, editor, *Advances in Cryptology - CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 136–146. Springer-Verlag, 1994.

[74] E. Kushilevitz. Privacy and communication complexity. *SIAM Journal of Disc. Math.*, 5(2):273–284, 1992.

[75] K. M. Martin. *Discrete Structures in the Theory of Secret Sharing.* PhD thesis, University of London, 1991.

[76] T. Matsumoto and H. Imai. On the key predistribution systems: A practical solution to the key distribution problem. In C. Pomerance, editor, *Advances in Cryptology – CRYPTO 87*, volume 293 of *Lecture Notes in Computer Science*, pages 185–193. Springer-Verlag, 1988.

[77] U. M. Maurer. Secret Key Agreement by Public Discussion from Common Information. *IEEE Trans. on Information Theory*, 39(3):733–742, 1993.

[78] K. S. McCurley. A key distribution scheme based on factoring. *Journal of Cryptology*, 1:95–105, 1988.

[79] R. J. McEliece and D. V. Sarwate. On sharing secrets and Reed-Solomon codes. *Communications of the ACM*, 24:583–584, September 1981.

[80] R. C. Merkle. Secure communication over insecure channels. *CACM*, 21(4):294–299, 1978.

[81] C. J. Mitchell and F. C. Piper. Key storage in secure networks. *Discrete Applied Mathematics*, 21(3):215–228, 1988.

[82] R. Molva, G. Tsudik, E. Van Herreweghen, and S. Zatti. *KryptoKnight* authentication and key distribution system. In *1992 European Symposium on Research in Computer Security*, pages 155–174, 1992.

[83] K. Mulmuley. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. *Combinatorica*, 7:101–104, 1987.

[84] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7:105–114, 1987.

[85] M. Naor and A. Shamir. Visual cryptography. In A. De Santis, editor, *Advances in Cryptology – Eurocrypt '94*, volume 950 of *Lecture Notes in Computer Science*, pages 1–12. Springer-Verlag, 1995.

[86] M. Naor and A. Wool. Access control and signatures via quorum secret sharing, 1995.

[87] E. I. Nečiporuk. A Boolean function. *Doklady of the Academy of Sciences of the USSR (in Russian)*, 169(4):765–766, 1966. English translation in *Soviet Mathematics Doklady* 7:4, pages 999-1000.

[88] E. I. Nečiporuk. On a Boolean matrix. *Problemy Kibernet.(in Russian)*, 21(4):237–240, 1969. English translation in *Systems Theory Res.*, 21 (1971) 236–239.

[89] B. B. Neuman and T. Ts'o. Kerberos: An authentication service for computer networks. *IEEE communications*, 32(9):33–38, 1994.

[90] E. Okamoto and K. Tanaka. Key distribution system based on identification information. *IEEE Journal on Selected Areas in Communications*, 7(4):481–485, 1989.

[91] N. Pippenger. On another Boolean matrix. *Theoretical Computer Science*, 11:49–56, 1980.

[92] K. A. S. Quinn. Some constructions for key distribution patterns. *Designs, Codes and Cryptography*, 4(2):177–191, 1994.

[93] M. O. Rabin. Randomized byzantine generals. In *Proceeding 24th Annual Symposium on the Foundations of Computer Science*, pages 403–409. IEEE, 1983.

[94] A. A. Razborov. A lower bound on the monotone network complexity of the logical permanent. *Mat. Zametki*, 37(6):887–900, 1985. In Russian, English translation in: *Math. Notes*, 37:485–493, 1985.

[95] A. A. Razborov. Lower bounds on monotone complexity of some Boolean functions. *Dokl. Ak. Nauk. SSSR*, 281:798–801, 1985. In Russian, English translation in: *Sov. Math. Dokl.*, 31:354–357, 1985.

[96] A.A. Razborov. On the method of approximation. In *Proceeding 21st Annual Symposium on the Theory of Computing*, pages 167–176. ACM, 1989.

[97] A.A. Razborov. Lower bounds for deterministic and nondeterministic branching programs, 1994.

[98] R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. In R. A. DeMillo, D. P. Dobkin, A. K. Jones, and R. J. Lipton, editors, *Foundations of Secure Computation*, pages 169–179. Academic Press, 1978.

[99] R. A. Rueppel and P. C. van Oorschoot. Modern key agreement techniques. *Computer Communications*, 17(7):458–465, 1994.

[100] P. D. Seymour. On secret-sharing matroids. *J. of Combinatorial Theory, Series B*, 56:69–73, 1992.

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

[102] C. E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, 1949.

[103] G. J. Simmons. How to (really) share a secret. In S. Goldwasser, editor, *Advances in Cryptology - CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 390–448. Springer-Verlag, 1990.

[104] G. J. Simmons. An introduction to shared secret and/or shared control and their application. In G. J. Simmons, editor, *Contemporary Cryptology, The Science of Information Integrity*, pages 441–497. IEEE Press, 1991.

[105] G. J. Simmons, W. Jackson, and K. M. Martin. The geometry of shared secret schemes. *Bulletin of the ICA*, 1:71–88, 1991.

[106] D. R. Stinson. An explication of secret sharing schemes. *Designs, Codes and Cryptography*, 2:357–390, 1992.

[107] D. R. Stinson. New general lower bounds on the information rate of secret sharing schemes. In E. F. Brickell, editor, *Advances in Cryptology - CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 168–182. Springer-Verlag, 1993.

[108] D. R. Stinson. Decomposition construction for secret sharing schemes. *IEEE Trans. on Information Theory*, 40(1):118–125, 1994.

[109] D. R. Stinson and S. A. Vanstone. A combinatorial approach to threshold schemes. *SIAM J. Disc. Math.*, 1(2):230–236, 1988.

[110] E. Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988.

[111] A. Wigderson. The fusion method for lower bounds in circuit complexity. In *Bolyai Society Mathematical Studies, Combinatorics, Paul Erdős is Eighty*, volume 1, pages 453–467, Keszthely (Hungary), 1993.

[112] A. Wigderson. $NL/poly \subseteq \oplus L/poly$. In *Proceeding of the 9th Annual Structure in Complexity Theory*, pages 59–62, 1994.

[113] Y. Yacobi and Z. Shmueley. On key distribution systems. In G. Brassard, editor, *Advances in Cryptology - CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 344–355. Springer-Verlag, 1990.

[114] A.C.C. Yao. unpublished manuscript.

Appendix A

Information Theory Definitions

We describe here the information-theoretic definitions and results used in Section 6.4 and in Section 7.5. For further details the reader may refer to [42, 45, 60]. Let X be a random variable distributed according to some probability distribution p. The *entropy* of X is defined as:

$$\mathrm{H}(X) \triangleq \sum_{x,p(x)>0} p(x) \log \frac{1}{p(x)} \ .$$

where the logarithm (here and elsewhere) is taken to the base two. Informally, the entropy measures the uncertainty of the random variable. The *conditional entropy* of X given Y is defined as:

$$\mathrm{H}(X|Y) \triangleq \sum_{y} p(y) \sum_{x,p(x|y)>0} p(x|y) \log \frac{1}{p(x|y)} \ .$$

The other two quantities defined in this appendix can be defined in terms of the above. The *mutual information* between X and Y is defined as:

$$\mathrm{I}(X;Y) \triangleq \mathrm{H}(X) - \mathrm{H}(X|Y) \ .$$

The *mutual information* between X and Y given Z is defined as:

$$\mathrm{I}(X;Y|Z) \triangleq \mathrm{H}(X|Z) - \mathrm{H}(X|YZ) \ .$$

The following can be easily verified.

1. $\mathrm{H}(X) \geq 0$ for all X with equality if and only if X is deterministic.
2. If X obtains n possible values then $\mathrm{H}(X) \leq \log n$ with equality if and only if X is uniformly distributed over these n values.
3. $0 \leq \mathrm{H}(X|Y) \leq \mathrm{H}(X)$ for all X and Y. Therefore, $0 \leq \mathrm{I}(X;Y) \leq H(X)$.
4. $\mathrm{H}(XY) = \mathrm{H}(X) + \mathrm{H}(Y|X)$.

5. $\mathrm{I}(X;Y) = \mathrm{I}(Y;X)$ for all X and Y.

6. $\mathrm{I}(X;Y|Z)$ can be larger, smaller, or the same as $\mathrm{I}(X;Y)$.

7. The random variables X and Y are independent if and only if $\mathrm{H}(X|Y) = \mathrm{H}(X)$.

Appendix B

Definition of Private Computations

We define *private protocols*, which are used in Chapter 8 for reconstructing functions of the secrets securely. The protocol is carried out in a system with secure private channels as defined in Definition 3.2 and Definition 3.3. We denote by M_T the communication received by a coalition T during the execution of the protocol (M_T is a random variable). We say that a coalition T *does not gain any additional information* (other than what follows from its input and the function value) from the execution of a randomized protocol F, which computes f, if the following holds: For every two inputs $\vec{x}, \vec{y}$ that agree in their T entries (i.e. $\forall i \in T : x_i = y_i$) and satisfy $f(\vec{x}) = f(\vec{y})$, and for every choice of random inputs $\{r_i\}_{i \in T}$, the messages passed between T and $\bar{T}$ are identically distributed. That is:

$$< M_T\left(\{x_i\}_{i\in T}, \{r_i\}_{i\in T}, \{x_i\}_{i\in \bar{T}}\right) > \; = \; < M_T\left(\{y_i\}_{i\in T}, \{r_i\}_{i\in T}, \{y_i\}_{i\in \bar{T}}\right) >$$

where the probability space is over all random inputs in $\bar{T}$, namely $\{r_i\}_{i\in\bar{T}}$. A protocol F for computing f is *private* if any coalition T of size at most n does not gain any additional information from the execution of the protocol. A function f is *private* if there exists a private protocol that computes it. It follows that if a reconstructing set uses a private protocol to reconstruct the secret, then the reconstruction is secure.

In our interactive secret sharing scheme with respect to the linear functions we use the private protocol of Benaloh [15] to compute the sum over GF(q) of the pieces. For the sake of completeness we describe this protocol in Fig. B.1.

The fact that the protocol described in Fig. B.1 computes the correct value is shown in the following equation.

$$\text{sum} = \sum_{j=1}^{n} y_i = \sum_{j=1}^{n}\sum_{i=1}^{n} r_{i,j} = \sum_{i=1}^{n} \underbrace{\sum_{j=1}^{n} r_{i,j}}_{x_i} = \sum_{i=1}^{n} x_i \,.$$

It is not to hard to show that every coalition of at most $n-1$ parties does not gain any information that is not implied by the sum. We omit the details.

Private addition over GF(q)

Each P_i has an input $x_i \in \mathrm{GF}(q)$.
Each P_i chooses $n-1$ random inputs from input GF(q) denoted $r_{i,1}, r_{i,2}, \ldots, r_{i,n-1}$.
Each P_i computes $r_{i,n} \triangleq x_i - \sum_{j=1}^{n} r_{i,j}$.
Each P_i sends $r_{i,j}$ to P_j (for every j).
Each P_j computes $y_j = \sum_{i=1}^{n} r_{i,j}$ and sends it to P_1.
Party P_1 computes sum $= \sum_{j=1}^{n} y_j$ and sends it to the other parties.

Figure B.1: Private addition over GF(q)
איור B.1: חישוב פרטי של סכום בשדה GF(q)