13

Crypto Topics and Applications II

Jennifer Seberry
University of Wollongong

Chris Charnes
Institute of Applied Physics and CASED Technical University Darmstadt

Josef Pieprzyk
Macquarie University

Rei Safavi-Naini
University of Calgary

13.1 Introduction

In this chapter we continue the exposition of crypto topics that was begun in the previous chapter. This chapter covers secret sharing, threshold cryptography, signature schemes, and finally quantum key distribution and quantum cryptography. As in the previous chapter, we have focused only on the essentials of each topic. We have selected in the bibliography a list of representative items, which can be consulted for further details.

First we give a synopsis of the topics that are discussed in this chapter.

Secret sharing is concerned with the problem of how to distribute a secret among a group of participating individuals, or entities, so that only predesignated collections of individuals are able to recreate the secret by collectively combining the parts of the secret that were allocated to them. There are numerous applications of secret-sharing schemes in practice. One example of secret sharing occurs in banking. For instance, the combination to a vault may be distributed in such a way that only specified collections of employees can open the vault by pooling their portions of the combination. In this way the authority to initiate an action, e.g., the opening of a bank vault, is divided for the purposes of providing security and for added functionality, such as auditing, if required.

Threshold cryptography is a relatively recently studied area of cryptography. It deals with situations where the authority to initiate or perform cryptographic operations is distributed among a group of individuals. Many of the standard operations of single-user cryptography have counterparts in threshold cryptography.

Signature schemes deal with the problem of generating and verifying (electronic) signatures for documents. A subclass of signature schemes is concerned with the shared-generation and the shared-verification of signatures, where a collaborating group of individuals are required to perform these actions.

A new paradigm of security has recently been introduced into cryptography with the emergence of the ideas of quantum key distribution and quantum cryptography. While classical cryptography employs various mathematical techniques to restrict eavesdroppers from learning the contents of encrypted messages, in quantum cryptography the information is protected by the laws of physics.

13.2 Secret Sharing

13.2.1 Introduction

Secret sharing is concerned with the problem of distributing a secret among a group of participating individuals, or entities, so that only predesignated collections of individuals are able to recreate the secret by collectively combining their shares of the secret.

The earliest and the most widely studied type of secret-sharing schemes are called (t, n)-threshold schemes. In these schemes the access structure—a precise specification of the participants authorized to recreate the secret, comprises all possible t-element subsets of an n-element set.

The problem of realizing, i.e., implementing, secret-sharing schemes for threshold structures was solved independently by Blakley [17] and Shamir [92] in 1979. Shamir's solution is based on the property of polynomial interpolation in finite fields; Blakley formulated and solved the problem in terms of finite geometries.

In a (t, n)-threshold scheme, each of the n participants holds some shares (also called shadows) of the secret. The parameter $t \leq n$ is the threshold value. A fundamental property of a (t, n)-threshold scheme is that the secret can only be recreated if at least t shareholders combine their shares, but less than t shareholders cannot recreate the secret. The fact that the key can be recovered from any t of the shares makes threshold schemes very useful in key management. Threshold schemes can tolerate the invalidation of up to $n - t$ shares—the secret can still be recreated from the remaining intact shares.

Secret-sharing schemes are also used to control the authority to perform critical actions. For example, a bank vault can be opened only if say, any two out of three trusted employees of the bank agree to do so by combining their partial knowledge of the vault combination. In this case, even if any one of the three employees is not present at any given time the vault can still be opened, but no single employee has sufficient information about the combination to open the vault.

Secret-sharing schemes that do not reveal any information about the shared secret to unauthorized individuals are called perfect. This notion will be formally defined in Section 13.2.2. In this survey, we discuss both perfect and nonperfect schemes, as the latter schemes are proving to be useful in various secret-sharing applications.

Besides the (t, n)-threshold structures, more general access structures are encountered in the theory of secret sharing. These will be considered in Section 13.2.8. General access structures apply to situations where the trust status of the participants is not uniform. For example, in the bank scenario described earlier, it could be considered more secure to authorize either the bank manager or any two out of three senior employees to open the vault.

Since Blakley's and Shamir's papers have appeared, the study of secret sharing has developed into an active area of research in cryptography. A fundamental problem of the theory and practice of

secret sharing is the issue of how to implement secret-sharing schemes for arbitrary access structures. We shall discuss later some of the solutions to this problem. Simmons [96] gives numerous examples of practical situations, which require secret-sharing schemes. He also gives a detailed account of the geometric approach to secret sharing. Stinson's [102] survey is broader and more condensed.

Simmons [95] discusses secret-sharing schemes with extended capabilities. He argues that there are realistic applications in which schemes with extended capabilities are required. In our discussion, we will assume that there is a key distribution center (KDC) that is trusted unconditionally.

13.2.2 Models of Secret Sharing

A common model of secret sharing has two phases. In the initialization phase, a trusted entity—the dealer—distributes shares of a secret to the participants via secure means. In the reconstruction phase, the authorized participants submit their shares to a combiner, which reconstructs the secret on their behalf. It is assumed that the combiner is an algorithm, which only performs the task of reconstructing the secret. We denote the sets of all possible secrets and shares by $\mathcal{K}$ and $\mathcal{S}$ respectively; the set of participants in a scheme is denoted by $\mathcal{P}$. Secret-sharing schemes can be modeled by the information theory concept of entropy (cf. [52]). This approach was initiated by Karnin et al. [63] and developed further by Capocelli et al. [28].

A secret-sharing scheme is a collection of two algorithms. The first (the dealer) is a probabilistic mapping:

$$\mathcal{D} : \mathcal{K} \to \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_n$$

where $\mathcal{S}_i \subset \mathcal{S}$ $(i = 1, 2, \ldots, n)$ and $\mathcal{S}_i$ is a subset of shares which is used to generate a share for the participant $P_i \in \mathcal{P}$. The second (the combiner) is a function:

$$\mathcal{C} : \mathcal{S}_{i_1} \times \mathcal{S}_{i_2} \times \cdots \times \mathcal{S}_{i_t} \to \mathcal{K}$$

such that if the corresponding subset of participants $\{P_{i_1}, P_{i_2}, \ldots, P_{i_t}\}$ belongs to the access structure Γ, it produces the secret $K \in \mathcal{K}$, i.e.,

$$H\left(K \mid P_{i_1}, P_{i_2}, \ldots, P_{i_t}\right) = 0. \tag{13.1}$$

The combiner fails to recompute the secret if the subset of participants does not belong to the access structure Γ, i.e.,

$$H\left(K \mid S_l\right) \geq 0 \tag{13.2}$$

for $S_l = \{s_{i_1}, s_{i_2}, \ldots, s_{i_l}\}$ and $S_l \notin \Gamma$.

In Equation 13.1, $H(K \mid P_{i_1}, P_{i_2}, \ldots, P_{i_t})$ is calculated with respect to the shares of the participants. A secret-sharing scheme is called perfect if $H(K \mid S_l) = H(K)$ for any unauthorized subset of participants, i.e., not belonging to an access structure Γ (cf. Section 13.2.8).

The following result is proved by Karnin et al. [63]. A necessary condition for a perfect threshold scheme is that for each share s_i, the inequality $H(s_i) \geq H(K)$ holds.

Most of the secret-sharing schemes that we discuss satisfy this inequality, but we will also consider in Section 13.2.12 schemes that do not satisfy this inequality; these are called nonperfect schemes.

13.2.2.1 The Matrix Model

A matrix representation of perfect secret-sharing schemes was introduced by Brickell and Stinson [26]. The matrix model is often used in theoretical investigations of secret sharing.

In this model, a perfect secret-sharing scheme is formulated as a matrix M that is known by all the participants $\mathcal{P}$ in the scheme. The $|\mathcal{P}|+1$ columns of M are indexed as follows. The first column corresponds to the dealer $\mathcal{D}$, the remaining columns are indexed by the remaining participants in $\mathcal{P}$. Each row of M contains one of the possible keys K that is to be shared in column $\mathcal{D}$, and the shares of K are located in the remaining columns. When the dealer wants share K, a row r that has K in the $\mathcal{D}$-column is chosen uniformly and randomly. The dealer distributes the shares of K to each participant using the matrix M, i.e., participant P_j receives the entry $M_{r,j}$ as his share.

The general requirements of a perfect secret scheme translate into the following combinatorial conditions in the matrix model; see Stinson [102], and Blundo et al. [20]. Suppose that Γ is an access structure:

1. If $\mathcal{B} \in \Gamma$ and $M(r,P) = M(r',P)$ for all $P \in \mathcal{B}$, then $M(r,\mathcal{D}) = M(r',\mathcal{D})$.
2. If $\mathcal{B} \notin \Gamma$, then for every possible assignment f of shares to the participants in $\mathcal{B}$, say $f = (f_P : P \in \mathcal{B})$, a nonnegative integer $\lambda(f,\mathcal{B})$ exists such that

$$\left|\left\{r : M(r,P) = f_P \; \forall P \in \mathcal{B},\; M(r,D) = K\right\}\right| = \lambda(f,\mathcal{B})$$

   is independent of the value of K.

13.2.2.2 Information Rate

The information rate of secret-sharing schemes was studied by Brickell and Stinson [26]. It is a measure of the amount of information that the participants need to keep secret in a secret-sharing scheme. The information rate of a participant P_i in a secret-sharing scheme with $|S_i|$ shares is

$$\rho_i = \frac{\log_2 |\mathcal{K}|}{\log_2 |S_i|}.$$

The information rate of the scheme, denoted ρ, is defined to be the minimum of the ρ_i.

A proof of the fact that $\rho \leq 1$ is given by Stinson [102]. This result motivates the definition of ideal secret-sharing schemes.

A perfect secret-sharing scheme is called ideal if $\rho = 1$; that is, if the size of each participants share, measured in the number of bits, equals the size of the secret.

We now define another measure used to quantify the comparison between secret-sharing schemes (cf. Section 13.2.8).

$\rho^*(\Gamma)$ is the maximum value of ρ for any perfect secret-sharing scheme realizing the access structure Γ.

For any access structure it is desirable to implement a secret-sharing scheme with the information rate close to 1. This minimizes the amount of information that needs to be kept secret by the participants, which means that there is a greater chance of the scheme remaining secure. For example, a (t,n)-threshold scheme implemented by Shamir's method is ideal, but when the scheme is modified to prevent cheating as proposed by Tompa and Woll [107], it is no longer ideal (cf. Section 13.2.7).

13.2.3 Some Known Schemes

We now describe several well-known threshold secret-sharing schemes.

13.2.3.1 Blakley's Scheme

Blakley [17] implements threshold schemes using projective spaces over finite fields $GF(q)$. A projective space $PG(t,q)$ is obtained from the corresponding $(t+1)$-dimensional vector space

$V(t+1,q)$ by omitting the zero vector of $V(t+1,q)$, and identifying two vectors v and v', which satisfy the relation $v = \lambda v'$ for some nonzero λ in $GF(q)$. This equivalence relation partitions the vectors of $V(t+1,q)$ into equivalence classes, i.e., the lines through the origin of $V(t+1,q)$. These are the points of $PG(t,q)$, and there are $(q^t-1)/(q-1)$ such points. Similarly, each k-dimensional subspace of $V(t+1,q)$ corresponds to a $(k-1)$-dimensional subspace of $PG(t,q)$. And every point of $PG(t,q)$ lies on $(q^t-1)/(q-1)$ $(t-1)$-dimensional subspaces, which are called the hyperplanes of $PG(t,q)$.

To realize a (t,n)-threshold scheme, the secret is represented by a point p chosen randomly from $PG(t,q)$; each point p belongs to $(q^t-1)/(q-1)$ hyperplanes. The shares of the secret are the n hyperplanes, which are randomly selected and distributed to the participants. If q is sufficiently large and n is not too large, then the probability that any t of the hyperplanes intersect in some point other than p is close to zero; cf. Blakley [17]. Thus in general the secret can be recovered from any t of the n shares. The secret cannot be recovered from the knowledge of less than t hyperplanes, as these will intersect only in some subspace containing p. This scheme is not perfect, since a coalition of unauthorized insider participants has a greater chance of guessing the secret than an unauthorized group of outsider participants.

Blakley's geometric solution to the secret-sharing problem has grown into an active area of research. We will cover some of these developments in this survey.

13.2.3.2 Simmons' Scheme

Simmons formulates secret-sharing schemes in terms of affine spaces instead of projective spaces. The reasons for using affine spaces instead of projective spaces are explained by Simmons [96]. (There is a correspondence between projective spaces and affine spaces, cf. Beth et al. [14].) Briefly, an affine space $AG(n,q)$ consists of points—the vectors of $V(n,q)$, and a hierarchy of l-dimensional subspaces for $l \leq n$ and their cosets. These correspond to the equivalence classes in projective geometry mentioned earlier, and are called the flats of $AG(n,q)$. The equivalence classes of lines, planes, etc., of $AG(t,q)$ are the 1-dimensional, 2-dimensional, etc., flats. A hyperplane is a flat of codimension one. To realize a (t,n)-threshold scheme in $AG(t,q)$, the secret is represented by a point p chosen randomly from $AG(t,q)$, which lies on a publicly known line V_d (lines have q points). A hyperplane V_i of the indicator variety is selected so that V_i intersects V_d in a single point p. The shares of the secret are the subsets of points of V_i. An authorized subset of participants, which spans V_i, enables the reconstruction of the secret. If an unauthorized subset of participants attempts to reconstruct the secret, their shares will only span a flat that intersects V_d in the empty set. Thus they gain no information about the secret. The precise amount of information gained by the unauthorized participants about the secret can be expressed in terms of the defining parameters of $AG(n,q)$. These schemes are perfect. Simmons [96] gives a detailed explanation of the implementation of secret-sharing schemes using projective and affine spaces.

13.2.3.3 Shamir's Scheme

Shamir's [92] scheme realizes (t,n)-access structures using polynomial interpolation over finite fields. In his scheme, the secrets $\mathcal{S}$ belong to a prime power finite field $GF(q)$, which satisfies $q \geq n+1$. In the initialization phase, the dealer $\mathcal{D}$ chooses n distinct nonzero elements $\{x_1,\ldots,x_n\}$ from $GF(q)$ and allocates these to participants $\{P_1,\ldots,P_n\}$. This correspondence is publicly known, and has undesirable side effects if any of the participants are dishonest; see Section 13.2.7. However for now, we will assume that all the participants obey faithfully the protocol for reconstructing the secret.

Fix a random element of $GF(q)$ as the secret K. The shares of K are created using the following protocol:

1. $\mathcal{D}$ chooses $a_1, a_2, \ldots, a_{t-1}$ from $GF(q)$ randomly, uniformly, and independently.
2. Let $a(x)$ be a polynomial of degree at most $t-1$, defined as $a(x) = K + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}$.
3. The shares of the secret key are $y_i = a(x_i)$, for $1 \leq i \leq n$.

With the aforementioned data, if any t out of the n participants $\{x_{i_1}, \ldots, x_{i_t}\}$ combine their shares $\{y_{i_1}, \ldots, y_{i_t}\}$, then by Lagrangian interpolation, there is a unique polynomial of degree at most $(t-1)$ passing through the points $\{(x_{i_1}, y_{i_1}), \ldots, (x_{i_t}, y_{i_t})\}$. So the combined shares of the t participants can be used to recreate the polynomial $a(x)$, and hence the secret, which is $K = a(0)$.

The relation between the secret and the shares is given by Lagrange's interpolation formula:

$$K = \sum_{j=1}^{t} y_{i_j} b_j, \tag{13.3}$$

where the b_j are defined as

$$b_j = \prod_{\substack{1 \leq k \leq t, \\ k \neq j}} \frac{x_{i_k}}{x_{i_k} - x_{i_j}}.$$

Shamir's scheme is computationally efficient in terms of the computational effort required to create the shares and to recover the secret. Also the share size is optimal in an information theoretic sense, cf. Definition 13.2.2.

The reconstruction phase in Shamir's scheme can be also considered as a system of linear equations, which are defined by the shares K_i. If t shares are submitted to the combiner, the system of linear equations

$$y_{i_j} = K + a_1 x_{i_j} + a_2 x_{i_j}^2 + \cdots + a_{k-1} x_{i_j}^{k-1}, \quad j = 1, \ldots, t$$

can be solved for the unknowns $K, a_1, a_2, \ldots, a_{t-1}$ because the determinant of this system of equations is a nonsingular Vandermonde determinant (the $\{x_1, \ldots, x_n\}$ are pair-wise distinct). However, if $t-1$ participants try to reconstruct the secret, they face the problem of solving $t-1$ linear equations in t unknowns. This system of equations has one degree of freedom. Consequently, $t-1$ participants do not obtain any information about the secret, as K was selected uniformly and randomly from $GF(q)$. Shamir's system is perfect.

13.2.3.4 A (t,t) Threshold Scheme

Karnin et al. [63] describe a secret-sharing scheme which realizes (t, t)-access structures. The interest in such schemes is that they can be used as the basis for other cryptographic constructions.

In their scheme, the set of secrets $\mathcal{S}$ is the ring of residue classes Z_m, where m is any integer. (In applications m is large.) The secret K is shared using the following algorithm:

1. $\mathcal{D}$ secretly chooses randomly, uniformly, and independently $t-1$ elements $y_1, y_2, \ldots, y_{t-1}$ from Z_m; the y_t are defined as

$$y_t = K - \sum_{i=1}^{t-1} y_i \bmod m\,.$$

2. Participant P_i for $1 \leq i \leq t$ receives the share y_i from $\mathcal{D}$.

The above system is perfect as the following argument shows. The set of shares of $l < t$ participants attempting to reconstruct the secret either contains the share $y_t = K - \sum_{i=1}^{t-1} y_i \bmod m$, or not. In both cases the (unauthorized) participants lack the necessary information to determine K. Shamir's scheme with $t = n$ provides an alternative construction of (t, t)-threshold schemes, using the fields $GF(q)$ instead of Z_m.

13.2.4 Threshold Schemes and Discrete Logarithms

The discrete logarithm has been widely employed in the literature to transform threshold schemes into conditionally secure schemes with extra properties. This idea is exploited in the papers by Benaloh [4], Beth [15], Charnes et al. [32], Charnes and Pieprzyk [33], Lin and Harn [69], Langford [66], and Hwang and Chang [58].

It is a consequence of the linearity of Equation 13.3 that Shamir's scheme can be modified to obtain schemes having enhanced properties, such as disenrollment capability, in which shares from one or more participants can be made incapable of forming an updated secret. (The formal analysis of schemes with this property was given by Blakley et al. [18].) Let $a(x)$ be a polynomial and let $a(i)$ be the shares in Shamir's scheme. In the modified threshold scheme proposed in [32], $g^{a(0)}$ is the secret and the shares are $s_i = g^{c_i}$, $c_i = a(i)$. A generator g of the cyclic group of the field $GF(2^n)$ is chosen so that $2^n - 1$ is a Mersenne prime.

The modified (t, n)-threshold schemes can disenrol participants whose shares have been compromised either through loss or theft, and still maintain the original threshold level. In the event that some of the original shares are compromised, the KDC can issue using a public authenticated channel a new generator g' of the cyclic group of $GF(2^n)$. The shareholders can calculate their new shares $s_i{}'$ from the initial secret data according to

$$s_i{}' = g'^{c_i}.$$

Hwang and Chang [58] used a similar setting to obtain dynamic threshold schemes.

Threshold schemes with disenrollment capability, without the assumption of the intractability of the discrete logarithm problem, can be based on families of threshold schemes. The properties of these schemes are studied in a paper by Charnes et al. [31]; here we provide the basic definition.

A threshold scheme family (TSF) is defined by an $(m \times n)$ matrix of shares $[s_{i,j}]$, such that

1. Any row $(s_{i,1}, s_{i,2}, \ldots, s_{i,n})$ represents an instance of $TS_{r_i}(t_i, n)$, where $i = 1, \ldots, m$.
2. Any column $(s_{1,j}, s_{2,j}, \ldots, s_{m,j})$ represents an instance of $TS_{c_j}(t_j, m)$, where $j = 1, \ldots, n$.

A family of threshold schemes in which all rows and all columns are ideal schemes is called an ideal *threshold scheme family*, or ITS family for short. In these schemes it is possible to alter dynamically the threshold values by moving from one level of the matrix to another.

Lin and Harn [69], and Langford [66] use the discrete logarithm to transform Shamir's scheme into a conditionally secure scheme which does not require a trusted KDC. A similar approach is used by Langford [66] to obtain a threshold signature scheme. Beth [15] describes a protocol for verifiable secret sharing for general access structures based on geometric schemes. The discrete logarithm problem is used to encode the secret and the shares so that they can be publicly announced for verification purposes.

It should be noted that the definition of disenrollment used in [32] is not the same as that of Blakley et al. [18]. Blakley et al. established a lower bound on the number of bits required to encode the shares in schemes that can disenrol participants. Their bound shows that this number grows linearly with the number of participants that the scheme can disenrol. They also present two geometric (t, n)-threshold schemes, which meet this bound.

It is interesting to note that Benaloh [4] used the discrete logarithm to transform Shamir's scheme, but for a very different purpose. One of the properties of the discrete logarithm is that the sum of the discrete logarithms of the shares of a secret is equal to the discrete logarithm of the product of the shares of the secret. This property has an application in secret-ballot elections (cf. Benaloh [4]) where, in contrast with schemes earlier mentioned the discrete logarithm problem is required to be tractable.

The homomorphic property introduced by Benaloh [4] has prompted the question whether similar schemes can be set up in noncommutative groups—other than the additive and the multiplicative groups of finite fields. It is an open problem to find useful applications of homomorphic schemes in abelian groups.

13.2.5 Error Correcting Codes and Secret Sharing

McEliece and Sarwate [75] observed that Shamir's scheme is closely related to Reed–Solomon codes [76]. The advantage of this formulation is that the error correcting capabilities of the Reed–Solomon codes can be translated into desirable secret-sharing properties.

Let $(\alpha_0, \alpha_1, \ldots, \alpha_{q-1})$ be a fixed list of the nonzero elements of a finite field $GF(q)$ containing q elements. In a Reed–Solomon code, an information word $\mathbf{a} = (a_0, a_1, \ldots, a_{k-1})$, $a_i \in GF(q)$, is encoded into the codeword $\mathbf{D} = (D_1, D_2, \ldots, D_{r-1})$, where $D_i = \sum_{j=0}^{k-1} a_j {\alpha_i}^j$. In this formulation, the secret is $a_0 = -\sum_{i=1}^{r-1} D_i$ and the shares distributed to the participants are the D_i.

In the above formulation of threshold schemes, algorithms, such as the errors-and-erasures decoding algorithm, can be used to correct t out of s shares where $s - 2t \geq k$ in a (k, n)-threshold scheme, if for some reason these shares were corrupted. The algorithm will also locate which invalid shares D_i were submitted, either as a result of deliberate tampering or as a result of storage degradation.

Karnin et al. [63] realize threshold schemes using linear codes. Massey [71] introduced the concept of minimal codewords, and proved that the access structure of a secret-sharing scheme based on a $[n, k]$ linear code is determined by the minimal codewords of the dual code. To realize a (t, n)-threshold scheme, a linear $[n + 1, t; q]$ code $\mathcal{C}$ over $GF(q)$ is selected. If G is the generator matrix of $\mathcal{C}$ and $s \in GF(q)$ is the secret, then the information vector $\mathbf{s} = (s_0, s_1, \ldots, s_{t-1})$ is any vector satisfying $s = \mathbf{s} \cdot \mathbf{g}^T$, where $\mathbf{g}^T$ is the first column vector of G. The codeword corresponding to $\mathbf{s}$ is $\mathbf{s}G = (t_0, t_1, \ldots, t_n)$. Each participant in the scheme receives t_i as its share and t_0 is the secret. To recover the secret, first the linear dependency between $\mathbf{g}$ and the other column vectors in the (public) generator matrix G is determined. If $\mathbf{g} = \sum x_j \mathbf{g}_j$ is the linear relation, the secret is given by $\sum x_j t_{i_j}$, where $\{t_{i_1}, t_{i_2}, \ldots, t_{i_t}\}$ is a set of t shares.

Renvall and Ding [84] consider the access structures of secret-sharing schemes based on linear codes as used by McEliece and Sarwate, and Karnin et al. They determine the access structures that arise from $[n + 1, k, n - k + 2]$ MDS codes—codes which achieve the singleton bound [76]. Bertilsson and Ingemarsson [13] use linear block codes to realize secret-sharing schemes for general access structures. Their algorithm takes a description of an access structure by a monotone Boolean formula Γ, and outputs the generator matrix of a linear code that realizes Γ.

13.2.6 Combinatorial Structures and Secret Sharing

There are various connections between combinatorial structures and secret sharing, cf. [14]. Stinson and Vanstone [105], and Schellenberg and Stinson [88] study threshold schemes based on combinatorial designs. Stinson [102] uses balanced incomplete blocks designs to obtain general bounds on the information rate ρ^* of schemes with access structure based on graphs (cf. Section 13.2.10).

Street [100] surveys defining sets for t-designs and critical sets for Latin squares, with the view of applying these concepts to multilevel secret-sharing schemes, in which a hierarchical structure can be imposed on the shares. To illustrate these methods, we give an example of a $(2, 3)$-threshold scheme based on a small Latin square, cf. Chaudhry and Seberry [36]. For an example of a scheme with a hierarchical share structure, cf. Street [100].

Let $(i, j; k)$ denote that the value k is in the position (i, j) of the Latin square:

$$L = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{pmatrix}.$$

The shares of the secret, which is L, are $S = \{(2, 1; 1), (3, 2; 1), (1, 3; 3)\}$.

More recently, critical sets in Room squares have been used to realize multilevel secret-sharing schemes, cf. Chaudhry and Seberry [36]. Some other approaches to multilevel schemes are considered in the papers by Beutelspacher [16] and Cooper et al. [38]. The schemes based on Latin and Room squares are examples of nonperfect schemes, which will be discussed in Section 13.2.12.

13.2.7 The Problem of Cheaters

So far we have assumed that the participants in a secret-sharing scheme are honest and obey the reconstruction protocol. However, there are conceivable situations where a dishonest clique of participants (assuming an honest KDC) may attempt to defraud the honest participants by altering the shares they were issued.

In the McEliece and Sarwate formulation of Shamir's scheme, invalid shares can be identified. Schemes with this capability are said to have the cheater identification property. A weaker capability ascertains that invalid shares were submitted in the reconstruction phase without necessarily locating the source of these shares; this is called cheater detection.

Tompa and Woll [107] show that the public knowledge of the abscissae in Shamir's scheme allows a clique of dishonest participants to modify their shares resulting in the recreation of an invalid secret K'. Suppose that participants $i_1, i_2, \ldots, i_t$ agree to pool their shares in order to recreate the secret. A dishonest participant, say i_1, can determine a polynomial $\Delta(x)$ of degree at most $t-1$ from $\Delta(0) = -1$ and $\Delta(i_2) = \Delta(i_3) = \cdots = \Delta(i_t) = 0$ using Lagrangian interpolation. Instead of the share originally issued by the dealer, the cheater submits the modified share $a(i_1) + \Delta(i_1)$. Lagrangian interpolation of points using the modified share will result in the polynomial $a(x) + \Delta(x)$ being recreated, instead of the intended polynomial $a(x)$. Now the constant term is $a(0) + \Delta(0) = K - 1$, a legal but incorrect secret. The honest participants believe that the secret is $K - 1$, while the cheater will privately recover the correct secret since $K = (K - 1) + 1$.

To prevent this type of cheating, Tompa and Woll define the shares in their modified scheme to be: $(x_1, d_1), (x_2, d_2), \ldots, (x_n, d_n)$. The dealer chooses randomly and uniformly a permutation $(x_1, x_2, \ldots, x_n)$ of n distinct elements from $\{1, 2, \ldots, q-1\}$, and $d_i = a(x_i)$. The modified scheme resists the aforementioned attack for up to $t - 1$ cheaters. The expected running time of the scheme is polynomial in $k, n, \log s$, and $\log(1/\epsilon)$, where ϵ is a designated security parameter of the scheme and the secret k is chosen from $\{0, 1, \ldots, s-1\}$. The overhead is that the participants need to keep secure two shares instead of the usual single share.

Using the error correcting code approach to secret sharing [75], Ogata and Kurosawa [81] formulated the problem of cheaters in secret-sharing schemes by introducing d_{cheat}. This is more appropriate than the minimum Hamming distance $d_{\min}$ of the related error correcting code in situations where only the correct secret needs to be recovered (as opposed to identifying the cheaters in the scheme). For general access structures Γ (see the following text), Ogata and Kurosawa prove

that $d_{\min} \leq d_{\text{cheat}} = n - \max_{B \notin \Gamma} |B|$. Secret-sharing schemes for which the two bounds coincide are called maximally distance separable, or (MDS) schemes.

Brickell and Stinson [25] modified Blakley's geometric (t, n)-scheme and obtained a scheme in which cheaters can be detected and identified. Blakley et al. [18] proved that this scheme can disenrol participants, cf. Section 13.2.4. To set up the scheme, the dealer performs certain computations, such as checking that the shadows are in general position. It is an open problem whether these computations can be done efficiently as the numbers of participants increase.

The problem of secret sharing without the usual assumptions about the honesty of the participants, or even the KDC, has been considered in the literature. For example, in verifiable secret sharing it is not assumed that the dealer is honest. This problem is studied by Chor et al. [37]. The problem is how to convince the participants in a (t, n)-threshold scheme that every subset of t shares of a share set $\{s_1, s_2, \ldots, s_n\}$ defines the same secret. This is called t-consistency. In Shamir's scheme, t-consistency is equivalent to the condition that interpolation on the points $(1, s_1), (2, s_2), \ldots, (n, s_n)$ yields a polynomial of degree at most $t - 1$. As application of homomorphic schemes, Benaloh [4] gives an interactive proof that Shamir's scheme is t-consistent.

13.2.8 General Access Structures

A complete discussion of secret sharing requires the notion of a general access structure.

Ito et al. [61] describe a method to realize secret-sharing schemes for general access structures. They observe that for most applications of secret sharing it suffices to consider monotone access structures (MAS) defined as follows.

Given a set $\mathcal{P}$ of n participants ($|\mathcal{P}| = n$), an MAS on $\mathcal{P}$ is a family of subsets $\mathcal{A} \subseteq 2^{\mathcal{P}}$ such that

$$\mathcal{A} \subseteq \mathcal{A}' \subseteq \mathcal{P} \Rightarrow \mathcal{A}' \in \mathcal{A} \tag{13.4}$$

The intersection $\mathcal{A}_1 \cap \mathcal{A}_2$ and the union $\mathcal{A}_1 \cup \mathcal{A}_2$ of two MASs is a MAS. If $\mathcal{A}$ is a MAS, then $2^{\mathcal{P}} \setminus \mathcal{A} = \bar{\mathcal{A}}$ is not a MAS. Any MAS can be expressed equivalently by a monotone Boolean function. Conversely, any Boolean expression without negations represents a MAS.

In view of the above observations, we consider the minimal authorized subsets of an access structure $\mathcal{A}$ on $\mathcal{P}$. A set $B \in \mathcal{A}$ is minimal authorized, if for each proper subset A of B, it is the case that $A \notin \mathcal{A}$. The set of minimal authorized subsets of $\mathcal{A}$ is called the basis. An access structure $\mathcal{A}$ is the unique closure of the basis, i.e., all subsets of $\mathcal{P}$ that are supersets of the basis elements.

Some examples of inequivalent access structures on four participants are given by the following monotone formulae: $\Gamma_1 = P_1P_2P_3 + P_1P_2P_4 + P_1P_3P_4 + P_2P_3P_4$ – a $(3, 4)$-threshold scheme; $\Gamma_2 = P_1P_2 + P_3P_4$; $\Gamma_4 = P_1P_2 + P_2P_3 + P_3P_4$. In these formulae, the P_i's represent the participants in the scheme (sometimes the literals A, B, etc., are used). The authorized subsets in the access structure are specified precisely by these formulae. For example, Γ_2 stipulates that either P_1 AND P_2 OR P_3 AND P_4 are the authorized subsets. (It is known that no threshold scheme can realize the access structure defined by Γ_2. For a proof, cf. Benaloh and Leichter [7].)

The inequivalent access structures on three and four participants, and the information rates of secret-sharing schemes realizing these structures are given by Simmons et al. [97] and Stinson [102]. The information rates of all inequivalent access structures on five participants are given in [73]. It should be remarked that a practical examination of access structures is probably limited to five participants, since the number of equivalence classes of monotone Boolean formulae becomes too great to consider for more than five participants. However, Martin and Jackson [73] provide inductive methods using which the information rates of an access structure Γ is related to the information rates of smaller access structures that are "embedded" in Γ.

Secret-sharing schemes for nonmonotone access structures have also been investigated, cf. Simmons [96].

13.2.9 Realizing General Access Structures

Ito et al. [61] were the first to show how to realize secret-sharing schemes for general access structures. Benaloh and Leichter [7] simplified the method of Ito et al.

They show that any monotone access structure can be recognized by a monotone Boolean circuit. In a monotone circuit, each variable corresponds to an element of $\mathcal{P}$. The circuit outputs a true value only when the set of variables which take a true value correspond to an authorized subset of $\mathcal{P}$, i.e., belongs to the access structure. Monotone circuits are described by Boolean formulae which involve only AND and OR operators. Using Benaloh and Leichter's method, one can realize any access structure as a composite of subsecrets. The subsecrets are shared across AND gates by (t, t)-threshold schemes for appropriate t, and all the inputs to the OR gates have the same value.

Simmons et al. [97] showed how cumulative arrays, first studied by Ito et al. [61], can be used to realize geometric secret-sharing schemes for general access structures.

A cumulative array $C_{\mathcal{A}} = (\mathcal{S}, f)_{\mathcal{A}}$ for the access structure $\mathcal{A}$ is a pair comprising of the share set $\mathcal{S} = \{s_1, s_2, \ldots\}$, and the dealer function $f : \mathcal{P} \to 2^{\mathcal{S}}$ that assigns subset of shares to each participant.

As an example, consider the following access structure:

$$\mathcal{A} = \text{closure } \{\{P_1, P_2\}, \{P_2, P_3\}, \{P_3, P_4\}, \{P_1, P_4\}\},$$

where $\mathcal{P} = \{P_1, P_2, P_3, P_4\}$. Let $\mathcal{S} = \{s_1, s_2\}$. A cumulative array for this access structure is $f(P_1) = s_1$, $f(P_2) = s_2, f(P_3) = s_1$, and $f(P_4) = s_2$.

Perfect geometric secret-sharing schemes are obtained from cumulative arrays as follows. Choose a projective space $V_i = PG(m-1, q)$, where m is the number of columns in the cumulative array. In V_i, let $\{s_i, \ldots, s_m, K\}$ be $m+1$ points, such that no m points lie on a hyperplane of V_i — the points are in general position. A domain variety V_d is chosen so that $V_i \bigcap V_d = \{K\}$. The set of shares in the geometric scheme is $\{s_i, \ldots, s_m\}$ and K is the secret. The shares are distributed using the cumulative array: participant P_i receives share s_j if and only if the (i, j) entry of the array is one. Note that it could be difficult to verify the general position hypothesis for large m, cf. Brickell and Stinson [25]. Jackson and Martin [62] show that any geometric secret-sharing scheme realizing an access structure is "contained" in the cumulative array, which realizes the access structure.

For any access structure $\mathcal{A}$ on the set $\mathcal{P}$, there is a unique minimal cumulative array. Thus to implement geometric secret-sharing schemes with the minimal number of shares, we need to consider only minimal cumulative arrays. It remains only to have a means by which the minimal cumulative array can be calculated given an arbitrary monotone Boolean function Γ. Such a method was first given by Simmons et al. [97]. It relies on minimizing the Boolean expression, which results when the AND and OR operators in Γ are exchanged.

An alternative method for calculating minimal cumulative arrays is described by Charnes and Pieprzyk [34]. This method has the advantage that the complete truth table of Γ is not required for some Γ, thereby avoiding an exponential time computation. For general Boolean expressions, as the number of variables increases the time complexity of the aforementioned method and that of [97] are the same.

To describe the method of [34], we require the following.

The representative matrix M_Γ of a monotone Boolean function $\Gamma(P_1, P_2, \ldots, P_n)$, expressed as a disjunctive sum of r products of n variables, is an $n \times r$ matrix with rows indexed by the P_i and columns by the product terms of the P_i [34]. The (i, j)-entry is one if P_i occurs in the jth product,

and is zero otherwise. For example, if $\Gamma = P_1P_2 + P_2P_3 + P_3P_4$, then M_Γ is the following matrix:

	P_1P_2	P_2P_3	P_3P_4
P_1	1	0	0
P_2	1	1	0
P_3	0	1	1
P_4	0	0	1

Suppose that $\Gamma(P_1, P_2, \ldots, P_n)$ is a monotone formula expressed in minimal disjunctive form, i.e., a disjunctive sum of products of the P_i and no product term is contained in any other representative set. Let M_Γ be its representative matrix.

[34] A subset $\{P_l, P_m, \ldots\}$ of the variables of $\Gamma(P_1, P_2, \ldots, P_n)$ is a relation *set* if $P_lP_m \ldots$ is represented in M_Γ by the all ones vector.

In the previous representative matrix $\{P_1, P_3\}$, $\{P_2, P_3\}$, and $\{P_2, P_4\}$ are the minimal representative sets, i.e., not contained in any other representative set. The Boolean formula derived from these sets is $P_1P_3 + P_2P_3 + P_2P_4$.

Let $\Gamma(P_1, P_2, \ldots, P_n)$ be a monotone formula and M_Γ its representative matrix [34]. Let $\mathcal{R}$ be the collection of minimal relation sets of M_Γ. Then the representative matrix whose rows are indexed by the variables P_i and columns by product terms derived from $\mathcal{R}$ is the minimal cumulative array for $\mathcal{A}$.

Thus, using the above theorem the matrix

	P_1P_3	P_2P_3	P_2P_4
P_1	1	0	0
P_2	0	1	1
P_3	1	1	0
P_4	0	0	1

is the minimal cumulative array for $\Gamma = P_1P_2 + P_2P_3 + P_3P_4$. To realize Γ as a geometric scheme, we require a projective space $V_i = PG(2, q)$. The secret $K \in V_d$, and the shares $\{s_1, s_2, s_3\}$ are points chosen in general position in V_i. The cumulative array specifies the distribution of the shares: P_1 receives share $\{s_1\}$; P_2 receives shares $\{s_2, s_3\}$; P_3 receives shares $\{s_1, s_2\}$; P_4 receives share $\{s_3\}$. It can be easily verified that only the authorized subsets of participants can recreate the secret, e.g., the combined shares of P_1 and P_2 span V_i, hence these participants can recover the secret as $V_i \cap V_d = \{K\}$. But unauthorized participants, e.g., P_1 and P_3, cannot recover the secret.

An algorithm for calculating cumulative arrays, based on Theorem 13.2.9, is given in [34]. This algorithm is efficient for those Γ for which M_Γ contains columns with many zeros. Thus in the previous example, the combinations $\{P_1, P_2\}$, $\{P_1, P_4\}$, and $\{P_3, P_4\}$ cannot produce relation sets and can be ignored. Further computational savings are obtained if the Boolean formula has symmetries; i.e., permutations of the participants that do not change Γ.

13.2.10 Ideal and Other Schemes

Brickell [23] gives a vector space construction for realizing ideal secret-sharing schemes for certain types of access structures, Γ. Let ϕ be a function

$$\phi : \mathcal{P} \cup \{\mathcal{D}\} \rightarrow GF(q)^d$$

with the property that $\phi(D)$ can be expressed as a linear combination of the vectors in $< \phi(P_i) : P_i \in B >$ if and only if B is an authorized subset, i.e., $B \in \Gamma$. Then, for any such ϕ, the distribution

rules (cf. Section 13.2.2.1) are for any vector $\mathbf{a} = (a_1, \ldots, s_d)$ in $GF(q)^d$, a distribution rule is given by the inner product of $\mathbf{a}$ and $\phi(x)$ for every $x \in \mathcal{P} \cup \{\mathcal{D}\}$. Under the previous conditions, the collection of distribution rules is an ideal secret-sharing scheme for Γ. A proof of this result can be found in a paper by Stinson [102].

Shamir's (t, n)-threshold scheme is an instance of the vector space construction. Access structures $\Gamma(G)$, whose basis is the edge set of certain undirected graphs, can also be realized as ideal schemes by this construction. In particular the access structure $\Gamma(G)$, where $G = (V, E)$ is a complete multigraph, can be realized as an ideal scheme. A proof of this is given by Stinson [102].

A relation between ideal secret sharing schemes and matroids was established by Brickell and Davenport [24]. The matroid theory counterpart of a minimal linearly dependent set of vectors in a vector space is called a circuit. A coordinatizable matroid is one that can be mapped into a vector space over a field in a way that preserves linear independence. Brickell and Davenport [24] prove the following theorem about coordinatizable matroids.

Suppose the connected matroid $\mathcal{M} = (X, \mathcal{I})$ is coordinatizable over a finite field [24]. Let $x \in X$ and let $\mathcal{P} = X \backslash \{x\}$. Then there exists an ideal scheme for the connected access structure having basis $\Gamma_0 = \{C \backslash \{x\} : x \in C \in \mathcal{C}\}$, where $\mathcal{C}$ denotes the set of circuits of $\mathcal{M}$.

Not all access structures can be realized as ideal secret-sharing schemes. This was first established by Benaloh and Leichter [7]. They proved that the access structure on four participants specified by the monotone formula $\Gamma = P_1P_2 + P_2P_3 + P_3P_4$ cannot be realized by an ideal scheme. The relation between the size of the shares and the secret for Γ was made precise by Capocelli et al. [28]. They proved the following information theoretic bound.

For the access structure $\Gamma =$ closure $\{\{P_1, P_2\}, \{P_2, P_3\}, \{P_3, P_4\}\}$ on four participants $\{P_1, P_2, P_3, P_4\}$, the inequality $H(P_2) + H(P_3) \geq 3H(K)$ holds for any secret-sharing scheme realizing Γ.

From this theorem, it follows that the information rate ρ of any secret-sharing scheme realizing Γ satisfies the bound $\rho \leq \frac{2}{3}$. Bounds are also derived by Capocelli et al. [28] for the maximum information rate ρ^* of access structures $\Gamma(G)$, where the graph G is a *path* P_n $(n \geq 3)$; a *cycle* C_n, $n \geq 6$, for n even and $n \geq 5$, for n odd; or any *tree* T_n.

13.2.11 Realizing Schemes Efficiently

In view of bounds on the information rates of secret-sharing schemes, it is natural to ask whether there exist schemes whose information rates equal the known bounds. For example, for $\Gamma = P_1P_2 + P_2P_3 + P_3P_4$ one is interested in the realizations of Γ with $\rho = \frac{2}{3}$.

Stinson [102] used a general method, called decomposition construction, to build larger schemes starting from smaller ideal schemes. In this method, the basis Γ_0 of an access structure is decomposed into smaller access structures, as $\Gamma_0 = \cup\Gamma_k$, where the Γ_k are the basis of the constituent access structures which can be realized as ideal schemes. From such decompositions of access structures, Stinson [102] derived a lower bound $\rho^*(\Gamma) \geq \ell/R$, where ℓ and R are two quantities defined in terms of the ideal decomposition of Γ_0. The decomposition construction and its precursor, the graph decomposition construction (cf. Blundo et al. [20]), can be formulated as linear programming problems in order to derive the best possible information rates that are obtainable using these constructions.

Other ways of realizing schemes with optimal or close to optimal information rates are considered by Charnes and Pieprzyk [35]. Their method combines multiple copies of cumulative arrays using the notion of composite shares—combinations of the ordinary shares in cumulative arrays. This procedure is stated as an algorithm that outputs a cumulative array with the best information rate. It is not clear how efficient this algorithm is as the number of participants increases. However, the optimal information rates for access structures on four participants given by Stinson [102] can be attained by combining cumulative arrays.

13.2.12 Nonperfect Schemes

It is known that in nonperfect secret sharing schemes the size of the shares is less that the size of the secret, i.e., $H(s_i) < H(K)$. Because of this inequality, a nonperfect scheme can be used to disperse a computer file to n sites, in such a way that the file can be recovered from its images that are held at any t of the sites for $t \leq n$. Moreover, this can be done so that the size of the images is less than the size of the original file resulting in an obvious saving of disk space. Making backups of computer files with this method provides insurance against the loss or the destruction of valuable data. For details, cf. Karnin et al. [63].

A formal analysis of nonperfect secret sharing schemes is given by Ogata et al. [80]. Their analysis characterizes, using information theory, secret-sharing schemes in which the participants not belonging to an access structure do gain some information about the secret. This possibility is precluded in perfect secret-sharing schemes.

Ogata et al. [80] define a nonperfect scheme in using a triple of access sets $(\Gamma_1, \Gamma_2, \Gamma_3)$, which partition the set of all subsets of the participants $\mathcal{P}$. Γ_1 is the family of access subsets, Γ_2 is the family of semiaccess subsets; and Γ_3 is the family of nonaccess subsets. The participants belonging to the semiaccess subsets are able to obtain some, but not complete information about the secret. The participants belonging to the nonaccess subsets gain no information about the secret.

The ramp schemes of Blakley and Meadows [19] are examples of nonperfect schemes where the access structure consists of semiaccess subsets. Another way of viewing ramp schemes is that the collective uncertainty about a secret gradually decreases as more participants join the collective.

Ogata et al. [80] prove a lower bound on the size of the shares in nonperfect schemes. They also characterize nonperfect schemes for which the size of the shares is $|K|/2$.

Ogata and Kurosawa [79] establish a general lower bound for the sizes of shares in nonperfect schemes. They show that there is an access hierarchy for which the size of the shares is strictly larger than this bound. It is in general a difficult problem to realize nonperfect secret-sharing schemes with the optimum share size, as in the case of perfect schemes.

13.3 Threshold Cryptography

There are circumstances in cryptography where an action requires to be executed by a group of people. For example, to transfer money from a bank a manager and a clerk need to concur. A bank vault can be opened only if three high-ranking bank employees cooperate. A ballistic missile can be launched only if two officers authorize the action.

Democratic groups usually exhibit a flat relational structure where every member has equal rights. On the other hand, in hierarchical groups, the privileges of group members depend on their position in the hierarchy. A member on the level $i - 1$ inherits all the privileges from the level i, as well as additional privileges specific to its position.

Unlike single-user cryptography, threshold or society-oriented cryptography allows groups to perform cryptographic operations, such as encryption, decryption, and signature. A trivial implementation of group-oriented cryptography can be achieved by concatenating secret-sharing schemes and a single user cryptosystem. This arrangement is usually unacceptable as the cooperating subgroup must first recover the cryptographic key. Having access to the key can compromise the system, as its use is not confined to the requested operation. Ideally, the cooperating participants should perform their private computations in one go. Their partial results are then sent to a so-called combiner who calculates the final result. Note that at no point is the secret key exposed.

A group-oriented cryptosystem is usually set up by a dealer who is a trusted authority. The dealer generates all the parameters, distributes elements via secure channels if the elements are secret, or broadcasts the parameters if they need not be protected. After setting up a group cryptosystem, the

dealer is no longer required, as all the necessary information has been deposited with the participants of the group cryptosystem.

If some participants want to cooperate to perform a cryptographic operation, they use a combiner to perform the final computations on behalf of the group. The final result is always correct if the participants belong to the access structure and follow the steps of the algorithm. The combiner fails if the participants do not belong to the access structure, or if the participants do not follow the algorithm (that is, they cheat). The combiner need not be trusted; it suffices to assume that it will perform some computations reliably but not necessarily all.

The access structure is the collection of all subsets of participants authorized to perform an action. An example is a (t, n)-threshold scheme, where any t out of n participants are authorized subsets $t \leq n$.

Threshold cryptography provides tools for groups to perform the following tasks:

- Threshold encryption—a group generates a valid cryptogram which can later be decrypted by a single receiver.
- Threshold decryption—a single sender generates a valid cryptogram that can be decrypted by a group.
- Threshold authentication—a group of senders agrees to co-authenticate the message so that the receiver can decide whether the message is authentic or not.
- Threshold signature (multisignature)—a group signs a message that is later validated by a single verifier.
- Threshold pseudorandom generation.

13.3.1 Threshold Encryption

Public-key cryptography can be used as a basis for simple group encryption. Assume that a receiver wants to have a communication channel from a group of n participants $\mathcal{P} = \{P_1, \ldots, P_n\}$. Further suppose that the receiver can decrypt a cryptogram only if all participants cooperate, i.e., a (n, n)-threshold encryption system. Group encryption works as follows.

Assume that the group and the receiver agree to use the RSA cryptosystem with the modulus $N = pq$. The receiver first computes a pair of keys: one for encryption e and the other for decryption d, where $e \times d \equiv 1 \bmod (p-1)(q-1)$. Both keys are secret. The factors p and q are known by the receiver only. The encryption key is communicated to the dealer (via a secure channel). The dealer selects $n-1$ shares e_i of the encryption key at random from the interval $[0, e/n]$. The last share is

$$e_n = e - \sum_{i=1}^{n-1} e_i.$$

Each share e_i is communicated to participant P_i via a secure channel $(i = 1, \ldots, n)$.

Now if the group wants to send a message m to the receiver, each participant P_i prepares a partial cryptogram $c_i \equiv m^{e_i} \bmod N$ $(i = 1, \ldots, n)$. After collecting n partial cryptograms, the receiver can recover the message $m \equiv (\prod_{i=1}^{n} c_i)^d \bmod N$. Note that the receiver also plays the role of a combiner. Moreover, the participants need not reconstruct the secret encryption key e and at no stage of decryption is the encryption key revealed—this is a characteristic feature of threshold cryptography.

Many existing secret-key algorithms, such as the DES [78], the LOKI [27], the FEAL [93], or the Russian GOST [99], are not homomorphic. These algorithms cannot be used for threshold encryption. The homomorphic property is necessary in order to generate shares of the key so that partial cryptograms can be combined into a cryptogram for the correct message, cf. [4].

Threshold encryption has not received a great deal of attention, perhaps because of its limited practical significance.

13.3.2 Threshold Decryption

Hwang [57] proposes a cryptosystem for group decryption based on the discrete logarithm problem. In his system, it is assumed that the sender knows the participants of the group. The sender encrypts the message using a predetermined (either private or public key) cryptosystem with a secret key known to the sender only. The sender then distributes the secret key among the group of intended receivers using Shamir's (t, n)-threshold scheme. Any t cooperating participants can recover the decryption key and decrypt the cryptogram. In Hwang's scheme, key distribution is based on the Diffie–Hellman [45] protocol. Thus the security of his scheme is equivalent to the security of the discrete logarithm problem. However, the main problem with the above solution is that the key can be recovered by a straightforward application of secret sharing. This violates the fundamental requirement that the decryption key must never be revealed to the group (or the combiner).

We consider now an implementation of a scheme for (t, n)-threshold decryption. The group decryption used here is based on the ElGamal public-key cryptosystem [47] and is described by Desmedt and Frankel [41].

The system is set up by the dealer $\mathcal{D}$ who first chooses a Galois field $GF(q)$ such that $q - 1$ is a Mersenne prime and $q = 2^{\ell}$. Further $\mathcal{D}$ selects a primitive element $g \in GF(q)$ and a nonzero random integer $s \in GF(q)$. The dealer computes $y = g^s \bmod q$ and publishes the triple (g, q, y) as the public parameters of the system. The dealer then uses Shamir's (t, n)-threshold scheme to distribute the secret s among the n shareholders in such a way that for any subset $\mathcal{B}$ of t participants, the secret $s = \sum_{P_i \in \mathcal{B}} s_i \bmod (q - 1)$ (all calculations are performed in $GF(q)$).

Suppose that user A wants to send a message $m \in GF(q)$ to the group. A first chooses at random an integer $k \in GF(q)$ and computes the cryptogram $c = (g^k, my^k)$ for the message m.

Assume that $\mathcal{B}$ is an authorized subset, so it contains at least t participants. The first stage of decryption is executed separately by each participant $P_i \in \mathcal{B}$. P_i takes the first part of the cryptogram and computes $(g^k)^{s_i} \bmod q$. The result is sent to the combiner, who computes $y^k = g^{ks} = \prod_{i \in \mathcal{B}} g^{ks_i}$, and decrypts (using the multiplicative inverse y^{-k}) the cryptogram

$$m \equiv my^k \times y^{-k} \bmod p.$$

Group decryption can also be based on a combination of the RSA cryptosystem [85] and Shamir's threshold scheme. The scheme described by Desmedt and Frankel [42] works as follows. The dealer D computes the modulus $N = pq$, where p, q are strong primes, i.e., $p = 2p' + 1$ and $q = 2q' + 1$ (p' and q' are large and distinct primes). The dealer selects at random an integer e such that e and $\lambda(N)$ are coprime ($\lambda(N)$ is the least common multiple of two integers $p - 1$ and $q - 1$, so $\lambda(N) = 2p'q'$). Next D publishes e and N as the public parameters of the system, but keeps p, q, and d secret (d satisfies the congruence $ed = 1 \bmod \lambda(N)$). It is clear that computing d is easy for the dealer who knows $\lambda(N)$, but is difficult—equivalent to the factoring of N—to someone who does not know $\lambda(N)$. The dealer then uses Shamir's scheme to distribute the secret $s = d - 1$ amongst n participants. The shares are denoted as s_i and any t cooperating participants (the set $\mathcal{B}$) can retrieve the secret. We have,

$$s = \sum_{i \in \mathcal{B}} s_i \bmod \lambda(N).$$

Group decryption of the cryptogram $c \equiv m^e \bmod N$ starts from individual computations. Each $P_i \in \mathcal{B}$ calculates a partial cryptogram $c^{s_i} \bmod N$. All the partial cryptograms are sent to the combiner who recovers the message

$$m = \prod_{i \in \mathcal{B}} c^{s_i} \times c \equiv c^{(\sum_{i \in \mathcal{B}} s_i + 1)} \equiv c^d \equiv \left(m^e\right)^d \bmod N.$$

Again the secret $s = d - 1$ is never exposed during the decryption.

Ghodosi et al. [53] proposed a solution to the problem of group decryption which does not require a dealer. It uses the RSA cryptosystem and Shamir's threshold scheme. The system works under the assumption that all participants from $\mathcal{P} = \{P_1, \ldots, P_n\}$ have their entries in a public registry (white pages). The registry provides the public parameters of a given participant. A participant P_i has N_i, e_i as its RSA entry in the registry, and this entry cannot be modified by an unauthorized person.

The sender first selects the group $\mathcal{P} = \{P_1, \ldots, P_n\}$. For the message m ($0 < m < \prod_{i=1}^{n} N_i$), the sender computes

$$m_i \equiv m \bmod N_i$$

for $i = 1, \ldots, n$. Next the sender selects at random a polynomial $f(x)$ of degree at most t over $GF(p)$, where $p < \min_i N_i$. Let

$$f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1}.$$

The sender computes $c_i = f(x_i)$ for public x_i, $k = f(0)$, $c_i^{e_i} \bmod N_i$, and $m_i^k \bmod N_i$ ($i = 1, \ldots, n$). Finally, the sender merges $c_i^{e_i} \bmod N_i$ into C_1 and $m_i^k \bmod N_i$ into C_2 using the Chinese Remainder Theorem. The sender broadcasts the tuple (N, p, t, C_1, C_2) as the cryptogram.

The participants check whether they are the intended recipients, for instance, by finding the gcd (N_i, N). Note that the sender can give the list of all participants instead of the modulus N. A participant P_i first recovers the pair $(c_i^{e_i} \bmod N_i)$ and $(m_i^k \bmod N_i)$ from C_1 and C_2, respectively. Using its secret key d_i, the participant retrieves c_i. The c_i are now broadcast so that each participant can reconstruct $f(x)$ and find $k = f(0)$. Note that none of the participants can cheat as it can be readily verified whether c_i' satisfies the congruence

$$c_i'^{e_i} \equiv C_1 \bmod N_i.$$

Knowing k, each participant finds the message $m_i \equiv C_2^{k^{-1}} \bmod N_i$. Although k is public, only participant P_i can find $k^{-1} \times k \bmod (p_i - 1)(q_i - 1)$ from his knowledge of the factorization of $N_i = p_i q_i$. Lastly, all the partial messages are communicated to the combiner who recovers the message m by the Chinese Remainder Theorem.

13.4 Signature Schemes

A signature scheme consists of two algorithms: signature generation and signature verification. Each of these algorithms can be collaboratively performed. A shared-generation scheme allows a group of signers to collaboratively sign a document. In a signature scheme with shared verification, the signature verification requires the collaboration of a group. We examine the two types of systems and note that the two can be combined if necessary.

13.4.1 Shared Generation Schemes

In these schemes, a signer group P of n participants has a public/private key pair. The private part is shared among members of the group such that each member has part of the private key that is not known to anyone else. The signature scheme is usually based on one of the well-known signature schemes, such as ElGamal, Schnorr, RSA, and Fiat–Shamir.

The group is created with an access structure that determines the authorized groups of signers. A special case of shared-generation schemes is the multisignature scheme, in which the collaboration of all members in P is necessary. Most systems proposed for shared generation are of the

multisignature type, or its generalization, (t, n)-threshold signature. In the latter type of signature, each subgroup p, $p \subset P$ of size t can generate the signature.

A shared-generation scheme can be sequential or simultaneous. In a sequential scheme, each member of the group signs the message and forwards it to the next group member. In some schemes, after the first signer the message is not readable and all subsequent signers must blindly sign the message. In a simultaneous scheme, each group member forms a partial signature that is sent to a combiner who forms the final signature.

There are a number of issues that differentiate shared-generation systems:

1. Mutually trusted party: a system may need a mutually trusted party who is usually active during the key generation phase; it chooses the group secret key and generates secrets for all group members. In systems without a trusted party, each signer produces his secret key and participates in a protocol with other signers to generate the group public key.
2. The security of most signatures schemes is based on the intractability of one of the following problems: discrete logarithm or integer factorization. Shared-generation schemes based on ElGamal and Schnorr signature schemes use the former, while those based on RSA and Fiat–Shamir use the latter.
3. Using many/few interactions for producing signature. The amount of interaction between the signers and the trusted third party varies in different schemes.

There are properties—some essential and some desirable—that a shared-generation scheme must satisfy. The essential properties are as follows:

A1 Signature generation must require the collaboration of all members of the authorized group and no signer in the group should be able to deny his signature. Verification must be possible by any outsider.

A2 An unauthorized group should not be able to forge the signature of an authorized group. It should not also be possible for an authorized group to forge the signature of another authorized group.

A3 No secret information should be derivable from the released group and partial signatures.

The desirable properties are as follows:

1. Each signer must have the same power and be able to see the message that he is signing.
2. The order of signing in a sequential scheme should not be fixed.
3. The size of the multisignature should be comparable to, preferably the same as, the size of the individual signature.

For a (t, n) threshold signature scheme, (A1) and (A2) reduce to

B1 From any t partial signature the group signature should be easily derivable.

B2 Knowledge of $t - 1$ or fewer partial signatures should not reduce the chance of forgery of an unauthorized group.

13.4.2 Constructions

The earliest proposals for shared-generation schemes are by Itakura and Nakamura [60] and by Boyd [21]. Boyd's scheme is a (n, n)-threshold group signature based on RSA, in which if $n > 2$ most participants must blindly sign the message.

13.4.2.1 Threshold RSA Signature

Desmedt and Frankel [42] construct a simultaneous threshold (t, n) RSA signature that requires a trusted third party to generate and distribute the group public key and the secret keys of the signers.

Their scheme works as follows. In the initialization stage, a trusted KDC (dealer) selects at random a polynomial of degree $t-1$: $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1}x^{t-1}$. The group secret key k is fixed as $a_0 = f(0)$. The dealer gives $y_i = f(x_i)$ to participant P_i, for each i, via a secure channel. The computations are performed in $Z_{\lambda(N)}$, where $\lambda = 2p'q'$ and $p = 2p' + 1, q = 2q' + 1$. To sign a message m $(0 \leq m < N)$, each participant $P_i \in B$ calculates their partial signature $s_i = m^{k_i} \bmod N$ and transmits the result to the combiner. The combiner computes the signature S of the message m according to the following equation:

$$S = m \times \prod_{P_i \in B} s_i = m \times \prod_{\substack{P_i \in B \\ i=1}}^{t} m^{k_i} = m \times m^{d-1} = m^d \bmod N.$$

The signature verification is similar to the conventional RSA signature scheme.

13.4.2.2 Threshold Signature Based on Discrete Logarithm

Ohta and Okamoto [82] propose a sequential multisignature scheme based on the Fiat–Shamir signature scheme. In their scheme, the order of signing is not restricted but the scheme requires a trusted center for key generation.

A variation of group signature is undeniable group signature, in which verification requires the collaboration of signers. The signature scheme has a "commitment phase" during which t group members work together to sign a message, and a "verification phase" during which all signers work together to prove the validity of the signature to an outsider. Harn and Yang [56] propose two (t, n)-threshold schemes with $t = 1$ and $t = n$. Their schemes do not require a trusted third party and the algorithm is based on the discrete logarithm problem.

Harn [55] proposes three simultaneous multisignature schemes based on the difficulty of discrete logarithm. Two of these schemes do not require a trusted third party. We briefly review one of the schemes. We use the notation of Harn [55].

Let KDC denote the key distribution center. The KDC selects

1. p, a large prime, in the range $2^{511} \leq p \leq 2^{512}$.
2. q, a prime divisor of $p - 1$.
3. $\{a_i, i = 0, \ldots, t-1\}$ and $f(x) = a_0 + a_1 x + \cdots + a_{t-1}x^{t-1} \pmod q$ where $0 < a_i < q$.
4. α, where $\alpha = h^{(p-1)/q} \pmod p > 1$. α is a generator with order q in GF(p); p, q and α are made public.

The KDC computes the group public key $y = \alpha^{f(0)} \bmod p$, where $f(0)$ is the group secret key. The KDC also computes public keys for all group members as

$$y_i = \alpha^{f(x_i)} \pmod p, \quad \text{for } i = 1, 2, \ldots, n$$

where $f(x_i) \bmod q$ is the share of participant i from the group secret key. (Note that, since α is a generator with order q in $GF(p)$, $\alpha^r \bmod p = \alpha^{r \bmod q}$, for any nonnegative integer r.)

In order to generate the group signature on a message m, each participant of a group B $(|B| \geq t)$ randomly selects an integer, $k_i \in [1, q-1]$, and computes a public value, $r_i = \alpha^{k_i} \bmod p$ and broadcasts r_i to all members in B. Knowing all the r_i $(i \in B)$, each member of the group B computes

$$r = \prod_{i \in B} r_i \pmod p.$$

Participant i computes his partial signature as

$$s_i = m' \times f(x_i) \times \left(\prod_{\substack{i,j \in B \\ ij/}} \frac{x_j}{(x_i - x_j)} \right) - k_i \times r \pmod{p}$$

where $H(m) = m'$ (H is a one-way and collision free hash function) and transmits (r_i, s_i) to a designated combiner.

Once the combiner receives the partial signature (r_i, s_i), it is verified using

$$y_i^{(m')^{\left(\prod_{\substack{i,j \in B \\ i \neq j}} \frac{x_j}{(x_i - x_j)} \right)}} = \alpha^{s_i} r_i^r \pmod{p}.$$

If all the partial signatures are verified, then the combiner calculates the group signature (r, s) on message m, where $s = \sum_{i \in B} s_i \pmod{q}$.

An outsider who receives the signature (r, s) on the message m can verify the validity of the signature using the check $y^{m'} = \alpha^s r^r \pmod{p}$. This check works because

$$f(0) = \sum_{i \in B} f(x_i) \prod_{\substack{i,j \in B \\ i \neq j}} \frac{x_j}{(x_i - x_j)} \pmod{q}$$

and thus,

$$y^{m'} = \left(\alpha^{f(0)} \right)^{m'} = \alpha^{\left(\sum_{i \in B} f(x_i) \prod_{\substack{i,j \in B \\ i \neq j}} \frac{x_j}{(x_i - x_j)} \right)^{m'}}$$

$$= \prod_{i \in B} y_i^{(m')^{\left(\prod_{\substack{i,j \in B \\ i \neq j}} \frac{x_j}{(x_i - x_j)} \right)}} = \prod_{i \in B} \left(r_i^r a_i^s \right) = r^r \alpha^s.$$

An interesting security problem in these schemes, as discussed by Desmedt and Frankel [42] and by Harn [55], is that if more than t signers collaborate they can find the secrets of the system with a high probability, and thus identify the rest of the shareholders. Possible solutions to this problem in the case of discrete logarithm-based schemes can be found in a paper by Li et al. [68].

A concept related to threshold signature is t-resilient digital signatures. In these schemes, n members of a group can collaboratively sign a message even if there are t dishonest members. Moreover no subset of t dishonest members can forge a signature.

Desmedt [40] shows that a t-resilient signature scheme with no trusted center can be constructed for any signature scheme using a general multiparty protocol. Cerecedo et al. [29] present efficient protocols for the shared generation of signatures based on the discrete logarithm problem, Schnorr's scheme, and variants of the ElGamal scheme. Their protocols are based on an efficient multiparty protocol for the shared computation of products and they do not need a trusted party. Park and Kurosawa [83] discuss a (t, n)-threshold scheme based on the discrete logarithm, more precisely a version of digital signature standard (DSS), which does not require multiplication and only uses linear combination for the combination of shares.

Chang and Liou [30] and Langford [66] propose other signature schemes based on the discrete logarithm problem.

13.4.3 Shared Verification of Signatures

Signature schemes with shared verification are not commonly found in the literature.

De Soete et al. [98] propose a system for the shared verification of signatures. But their system is not really a signature scheme in the sense that it does not produce a signature for every message. Each user has a secret that enables her/him to verify herself/himself to others. It requires at least two verifiers for the secret to be verified.

Laih and Yen [65] argue that in some cases it might be necessary to sign a message such that only a specified group of participants can verify the signed message. The main requirements of such schemes are

1. A can sign any message M for any specified group B.
2. Only the specified group can validate the signature of A. No other group, except B, can validate the signature of A on M.
3. B should not be able to forge A's signature on M for another user C, even if B and C conspire.
4. No one should be able to forge A's signature on another message M'.
5. If A disavows his signature, it must be possible for a third party to resolve the dispute between A and B.

The scheme proposed by Laih and Yen [65] is based on Harn's scheme, which is an efficient ElGamal type shared-generation scheme. In the proposed scheme a group of signers can create a digital shared-generation scheme for a specified group, who can collectively check the validity of the signature. The secret key of the users is chosen by the users themselves and each group has a public key for signature generation or verification. Since the private key of the verifiers is not known, dispute settlement by a third party requires an extra protocol between the third party and the verifiers.

13.5 Quantum Key Distribution—Quantum Cryptography

While classical cryptography employs various mathematical techniques to restrict eavesdroppers from learning the contents of encrypted messages, in quantum mechanics the information is protected by the laws of physics. In classical cryptography absolute security of information cannot be guaranteed. However on the quantum level there is a law called the Heisenberg uncertainty principle. This states that even the most refined measurement on a quantum object cannot reveal everything about the state of the object before the measurement. This is because the object may be altered by simply taking the measurement. The Heisenberg uncertainty principle and quantum entanglement can be exploited in a system of secure communication, often referred to as "quantum cryptography" [11]. Quantum cryptography provides the means for two parties to exchange a enciphering key over a private channel with complete security of communication.

There are least two main types of quantum cryptosystems for the key distribution:

- *BB-protocol*: Cryptosystems with encoding based on two nonsimultaneously measurable observables proposed by Wiesner [110] and Bennett and Brassard [10]
- *EPR-type*: Cryptosystems with encoding built upon quantum entanglement and the Bell Theorem proposed by Ekert [48]

A typical Quantum key distribution (QKD) protocol is comprised of the following stages:

1. Random number generation by Alice
2. Quantum communication
3. Sifting

4. Reconciliation
5. Estimation of Eve's partial information gain
6. Privacy amplification
7. Authentication of public messages
8. Key confirmation

For detailed explanations of these terms we refer the reader to [3], here we will give an informal explanation of some of these steps using the following simple example.

A *BB* quantum cryptosystem includes a transmitter, Alice, and a receiver, Bob. Alice may use the transmitter to send photons in one of the four polarizations: 0°, 45°, 90°, or 135°. Bob at the other end uses the receiver to measure the polarization. According to the laws of quantum mechanics, Bob's receiver can distinguish between rectilinear polarizations (0 and 90; basis I), or it can quickly be reconfigured to discriminate between diagonal polarizations (45 and 135; basis II); it can never, however, distinguish both types. The key distribution requires several steps.

Alice sends photons with one of the four polarizations which are chosen at random. For each incoming photon, Bob chooses at random the type of measurement: either the rectilinear type or the diagonal type. Bob records the results of the measurements but keeps them secret. Subsequently Bob publicly announces the type of measurement (but not the results) and Alice tells the receiver which measurements were correct—the measurement is done in the same basis as the preparation. Alice and Bob (the sender and the receiver) keep all cases in which Bob's measurements were of the correct type. These cases are then translated into bits (1's and 0's) and thereby become the key. An eavesdropper (Eve) is bound to introduce errors to this transmission because she does not know in advance the type of polarization of each photon and quantum mechanics does not allow her to acquire the sharp values of two nonsimultaneously measurable observables (here the rectilinear and diagonal polarizations).

The two legitimate users of the quantum channel, Alice and Bob, test for eavesdropping by revealing a random subset of the key bits and checking (in public) the error rate. Although they cannot prevent eavesdropping, they will never be fooled by Eve because any effort to tap the channel, however subtle and sophisticated, will be detected. Whenever they are not satisfied with the security of the channel they can try to set up the key distribution again. Privacy amplification is used to distill a secret key between Alice and Bob from these interactions, cf. Bennett et al. [12].

In our example, we will consider two types of measurement: we consider \ and / to be one type (diagonal) and | and − to be the other (rectilinear).

1. Alice's polarization	|	\	−	|	/	−	−	−	−	\	/
2. Bits Alice sent	0	0	1	0	0	1	1	1	0	1	1
3. Bob's polarization	|	\	|	\	|	|	/	|	−	\	−
4. Bits Bob registered	0	0	1	1	0	1	0	1	0	1	1
5. Alice states publicly whether Bob's polarization was correct or not	Yes	Yes	Yes	No	No	Yes	No	Yes	Yes	Yes	No
6. Alice's remaining bits	0	0	1	*x*	*x*	1	*x*	1	0	1	*x*
7. Bob's remaining bits *x* means discard	0	0	1	*x*	*x*	1	*x*	1	0	1	*x*
8. Alice and Bob compare bits chosen at random	0 OK	0	1	*x*	*x*	1	*x*	1 OK	0	1 OK	*x*
9. Alice's remaining bits	*x*	0	1	*x*	*x*	1	*x*	*x*	0	*x*	*x*
10. Bob's remaining bits	*x*	0	1	*x*	*x*	1	*x*	*x*	0	*x*	*x*
11. The key		0	1			1			0		

The basic idea of cryptosystems of *EPR*-type is as follows. A sequence of correlated particle pairs is generated, with one member of each pair being detected by Alice and the other by Bob (e.g., a pair prepared in a Bell state, whose polarizations are measured by Alice and Bob). To eavesdrop on this communication, Eve would have to detect a particle to read the signal, and retransmit it in order to conceal her presence. However, the act of detection of one particle of a pair destroys its quantum correlation with the other. Thus Alice and Bob can easily verify whether this has been done, without revealing the results of their own measurements, by communication over an open channel.

In the BB-protocol, Eve does not gain any information about the key material by passively monitoring Alice and Bob's public channel communications; however, it is essential that these messages are authenticated. Otherwise Eve could perform a man-in-the-middle attack in which he/she masquerades as Bob to Alice and as Alice to Bob, while forming separate keys with each. This scenario can be avoided if Alice and Bob append an authentication tag to their public messages, which is computed using a keyed hash function (cf. Wegman & Carter Construction, Section 12.5.2). Upon receiving a message, Alice and Bob can each verify that the received tag value matches the value computed from the message using the keyed hash function. This requires that Alice and Bob share a short initial key, which needs to be kept secret only for the duration of the transfer of Alice's photons to Bob. The QKD procedure produces large quantities of shared long-term secret bits, a few of which can be used to authenticate the next QKD session.

In the original proposal of Bennett and Brassard, the shared keys produced by quantum key distribution would be used directly for encryption as a one-time pad (encryption-mode QKD). With present day technology (c 2009), it is more practical to use QKD for the transfer or the generation of conventional symmetric cryptographic keys.

13.5.1 Practicalities—Quantum Cryptography

The field of quantum cryptography was pioneered by Wiesner [110]. Around 1970 at Columbia University in New York he showed how quantum effects could in theory be used to produce "quantum bank notes" that are immune to counterfeiting. The first feasible cryptosystems were proposed between 1982 and 1984 by the American physicist Charles H. Bennett of IBM's Thomas J. Watson Research Center in Yorktown Heights in the United States and by the Canadian expert in cryptography, Gilles Brassard from the University of Montreal [10]. In 1991 Artur Ekert [48] developed the idea along slightly different lines. This system is based on another aspect of quantum theory called entanglement.

By the early 1980s, the theory of secure quantum key distribution based on both the Heisenberg uncertainty principle and the quantum entanglement had evolved into a proposal for a testable system, though this was by no means easy to set up experimentally. The first apparatus, constructed by Bennett, Brassard and colleagues in 1989 at IBM's research center, was capable of transmitting a secret key over a distance of approximately 30 cm [9]. The practical use of this technology was established when the first demonstrations over optical fiber were implemented in the early 1990s.

Since then, other researchers have looked at systems based on correlations of another quantum property of light called phase. Phase is a measure of how far a photon has gone in its cycle of vibration. Information about the key is encoded in this property of phase instead of polarization. This has the advantage that with current technology, phase is easier to handle over a long distance. Since 1991 John Rarity and Paul Tapster of Britain's [49,106,108] Defense Research Agency in Malvern have been developing a system to increase the transmission distance. They have designed and tested an optical system good enough to transmit photons that stay correlated in phase over several hundred meters. With improvements in the technology of optical fibers and semiconductor photo detectors, which allowed better transmission and detection and thereby reducing background errors, the maximum transmission distance achieved by the mid-1990s was approximately 10 km.

In theory, cryptosystems based on entanglement should also allow quantum keys to be stored by storing photons without performing any measurements. At present, however, photons cannot be kept correlated longer than a small fraction of a second, so they are not a good medium for information storage. But a fraction of a second is long enough for a photon to cover a long distance, so photons are suitable for sending information and for key distribution.

Since the mid-1990s there have been important advances in quantum cryptography, such as

- Increase in transmission distance to about 150 km
- Implementations of practical systems (from the optical table to a 19″ rack)
- Development of commercial products

Current research in QKD focuses on implementations, which use weak laser pulses, single-photon sources, entangled pairs, and continuous variables. A comprehensive evaluation of the transmission distances that can be achieved by these technologies and their relative merits can be found in the Quantum Information and Technology Roadmapping Project [3]. The review papers [54,74,101] provide details of the current experimental implementations of QKD.

In 2006, IdQuantique (Switzerland) and Senetas (Australia) released commercially a hybrid quantum/classical encryption system. This system uses quantum key distribution to securely exchange encryption keys, which are used by a high speed classical network for encryption at speeds up to 10 Gbps. It was used in October 2007 to secure the transmission of federal election results in the State of Geneva. The system features AES 256 bit encryption, BB84 and SARG [87] protocols for QKD, HMAC-SHA-1 authentication for the classical link, and Wegmann & Carter for the QKD link. The cyryptographic keys are refreshed at the rate of 1 key/min up to four encryption appliances, and transmitted by a quantum link channel of length up to 80 km on single mode dark fiber.

13.5.2 Shor's Quantum Factoring Algorithm

Mathematicians have tried hard to solve the key distribution problem, and in the 1970s a clever mathematical discovery called "public-key" systems gave an elegant solution. Public-key cryptosystems avoid the key distribution problem, but unfortunately their security depends on unproven mathematical assumptions, such as the difficulty of factoring large integers (RSA, the most popular public key cryptosystem, gets its security from the difficulty of factoring large numbers). An enemy who knows your public key can in principle calculate your private key because the two keys are mathematically related; however, the difficulty of computing the private key from the respective public key is exactly that of factoring big integers.

Difficulty of factoring grows rapidly with the size, i.e., number of digits, of the number we want to factor. To see this, take a number N with ℓ decimal digits ($N \approx 10^{\ell}$) and try to factor it by dividing it by 2, 3, $\ldots$, $\sqrt{N}$ and checking the remainder. In the worst case, approximately $\sqrt{N} \approx 10^{\ell/2}$ divisions may be needed to solve the problem—an exponential increase as a function of ℓ. Now imagine computer capable of performing 10^{10} divisions per second. The computer can then factor any number N, using the trial division method, in about $\sqrt{N}/10^{10}$ s. Take a 100-digit number N, so that $N \approx 10^{100}$. The computer will factor this number in about 10^{40} s which is much longer than 10^{17} s—the estimated age of the universe!

It seems that factoring big numbers will remain beyond the capabilities of any realistic computing devices and unless mathematicians or computer scientists come up with an efficient factoring algorithm public-key cryptosystems will remain secure. Or will they? As it turns out we know that this is not the case; the classical, purely mathematical, theory of computation is not complete simply because it does not describe all physically possible computations. In particular, it does not describe computations that can be performed by quantum devices. Indeed, recent work in quantum computation shows that a quantum computer can factor much faster than any classical computer.

Quantum computers can compute faster because they can accept as input not just one number, but a coherent superposition of many different numbers, and subsequently perform a computation (a sequence of unitary operations) on all of these numbers simultaneously. This can be viewed as a massive parallel computation, but instead of having many processors working in parallel we have only one quantum processor performing a computation that affects all the components of the state vector. To see how it works let us describe Shor's factoring using a quantum computer composed of two quantum registers.

Consider two quantum registers, each register composed of a certain number of two-state quantum systems, which we call "qubits" (quantum bits). We take the first register and place it in a quantum superposition of all the possible integer numbers it can contain. This can be done by starting with all qubits in the $|0\rangle$ states and applying a simple unitary transformation to each qubit that creates a superposition of $|0\rangle$ and $|1\rangle$ states:

$$|0\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \tag{13.5}$$

Imagine a two-qubit register, for example. After this procedure the register will be in a superposition of all four numbers it can contain

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \tag{13.6}$$

where $|00\rangle$ is binary for 0, $|01\rangle$ is binary for 1, $|10\rangle$ is binary for 2, and finally $|11\rangle$ which is binary for 3.

Then we perform an arithmetical operation that takes advantage of quantum parallelism by computing the function $F_N(x)$ for each number x in the superposition. The values of $F_N(x)$ are placed in the second register so that after the computation the two registers become entangled:

$$\sum_x |x\rangle|F_N(x)\rangle \tag{13.7}$$

(we have ignored the normalization constant). Now we perform a measurement on the second register. The measurement yields a randomly selected value $F_N(k)$ for some k. The state of the first register immediately after the measurement, due to the periodicity of $F_N(k)$, is a coherent superposition of all states $|x\rangle$ such that $x = k,\ k + r,\ k + 2r, \ldots$, i.e., all x for which $F_N(x) = F_N(k)$. The value k is randomly selected by the measurement: therefore, the state of the first register is subsequently transformed via a unitary operation that sets any k to 0 (i.e., $|k\rangle$ becomes $|0\rangle$ plus a phase factor) and modifies the period from r to a multiple of $1/r$. This operation is known as the quantum Fourier transform. The first register is then ready for the final measurement and yields an integer, which is the best whole approximation of a multiple of $1/r$. From this result r, and subsequently factors of N, can be easily calculated (see the following text). The execution time of the quantum factoring algorithm can be estimated to grow as a quadratic function of ℓ, and numbers 100 decimal digits long can be factored in a fraction of a second!

When the first quantum factoring devices are built, the security of public-key cryptosystems will vanish. The mathematical solution to the key distribution problem is shattered by the power of quantum computation. Does it leave us without any means to protect our privacy? Fortunately quantum mechanics after destroying classical ciphers comes to rescue our privacy and offers its own solution to the key distribution problem.

The main reference for this brief account of quantum cryptanalysis is Shor's paper [94]. Ekert and Jozsa [50] give a comprehensive exposition of Shor's algorithm for factoring on a quantum computer, which includes some relevant background in number theory, computational complexity theory, and quantum computation as well as remarks about possible experimental realizations.

13.5.2.1 Factoring

Quantum factoring of an integer N is based on calculating the period of the function $F_N(x) = a^x \pmod N$. We form the increasing powers of a until they start to repeat with a period we denote r. Once r is known, the factors of N can be found using the Euclidean algorithm to find the greatest common divisor of $a^{r/2} \pm 1$ and N.

Suppose we want to factor 91. Let us take a number at random, say 28, and raise it to the powers 2, 3, ... After 12 iterations we find the number 28 repeated and so we use $r = 12$. Hence we want to find $\gcd(91, 28^6 \pm 1)$, which we find to be 1 and 13, respectively. From here we can factorize 91. Classically, calculating r is as difficult as trying to factor N and the execution time is exponential in the number of digits in N. Quantum computers can find r in time, which grows only as a quadratic function of the number of digits of N.

13.5.3 Practicalities—Quantum Computation

The idea of a quantum computer is simple, however, its realization is not. Quantum computers require a coherent, controlled evolution for a period of time, which is necessary to complete the computation. Many view this requirement as an insurmountable experimental problem; however, others believe that technological progress will sooner or later make such devices feasible. In an ordinary, classical computer, all the bits have a definite state at a given instant in time, say 0 1 1 0 0 0 1 0 1 0 .... However, in a quantum computer the state of the bits is described by a pure quantum state of the form

$$\Psi = a|0\,1\,1\,0\,0\,0\,1\,0\,1\ldots\rangle + b|1\,1\,1\,0\,0\,0\,0\,0\,1\ldots\rangle. \tag{13.8}$$

The coefficients $a, b, \ldots$ are complex numbers and the probability that the computer is in the state 0 1 1 0 0 0 1 0 1 ... is $|a|^2$, that it is in the state 1 1 1 0 0 0 0 0 1 ... is $|b|^2$, and so on. However, describing the state of the computer by the quantum state of 13.8 does not merely imply the ordinary uncertainties that we describe using probabilities. For instance, the phases of the complex coefficients $a, b, \ldots$ have genuine significance: they can describe interference between different states of the computer, which is very useful for quantum computation. The quantum state declares that the computer exists in all of its computational basis states simultaneously so long as that state is not measured; when we do choose to measure it, a particular computational basis state will be observed with the prescribed probability.

Since quantum computers have the potential to perform certain computational tasks more efficiently than their classical counterparts, there has been an intense international research effort aimed at realizing these devices. In early research on the synthesis of quantum logical circuits, DiVincenzo [46] showed how to construct the quantum analogue of the one-bit NOT, or inverter gate, with spectroscopic techniques that have been well known in physics for over 50 years. He established that the three-qubit AND operation can be performed using three XOR gates and four single-qubit rotations.

There are currently several promising approaches to the realization of a quantum computer: nuclear magnetic resonance (NMR), trapped ions, neutral atoms, cavity QED, optical, solid state, and superconducting devices. The potentials of these technologies are evaluated by the DiVincenzo criteria in the Quantum Information and Technology Roadmapping Project [2], which also gives medium- and long-term objectives for the realization of a quantum computer.

A prominent example of this technology is the Cirac Zoller proposal for a scalable quantum computer based on a string of trapped ions whose electronic states represent the quantum bits of information. In this scheme, quantum logical gates involving any subset of ions are realized by coupling the ions through their collective quantized motion. In 2003, the CNOT quantum gate according to the Cirac Zoller proposal was experimentally demonstrated by Schmidt-Kaler et al. [89]

In 2001, Vandersypen et al. [109] realized experimentally Shor's algorithm using liquid-state NMR techniques. In 2007, a linear optics realization of Shor's algorithm was demonstrated by Lanyon et al. [67]. Both teams chose the simplest instance of this algorithm, i.e., factorization of "$N = 15$". The linear optics experiment represents an essential step toward the full realization of Shor's algorithm and scalable linear optics quantum computation.

13.6 Further Information

As in the previous chapter we mention the conferences CRYPTO, EUROCRYPT, ASIACRYPT, AUSCRYPT, and conferences dealing with security such as ACISP. Information can also be found in Journals such as the *Communications of the ACM.* Quantum cryptography is also covered in the physics literature, e.g., *Europhysics Letters, Physical Review Letters,* and QIPC Strategic Report: Quantum Information Processing and Communication in Europe http://qist.ect.it/Reports/reports.htm

Acknowledgments

We thank Julian Fay of Senetas Australia, Gregoire Ribordy of id Quantique Switzerland and Professor Gernot Alber of IAP TUD Darmstadt for providing comments and suggestions which have greatly improved our exposition.

References

1. Van Assche, G., Cardinal, J., and Cerf, N.J., Reconciliation of a quantum-distributed gaussian key, *IEEE Transactions on Information Theory*, 50, 2004, arXiv:quant-ph/0107030.
2. ARDA, A quantum information and technology roadmapping project, Part 1: Quantum computation, LA-OR-04-1778, 2004, http://qist.lanl.gov/qcomp_map/shtml.
3. ARDA, A quantum information and technology roadmapping project, Part 2: Quantum cryptography, LA-OR-04-4085, 2004, http://qist.lanl.gov/qcrypt_map/shtml.
4. Benaloh, J.C., Secret sharing homomorphisms: Keeping shares of a secret secret. *Proceedings of Crypto'86, LNCS* Vol. 263, pp. 251–260, Springer-Verlag, Berlin, 1987.
5. Bellare, M. and Rogaway, P., Robust computational secret sharing and a unified account of classical secret sharing goals, http://eprint.iacr.org/2006/449.
6. Belenkiy, M. Disjunctive multi-level secret sharing, http://eprint.iacr.org/2008/018.
7. Benaloh, J. and Leichter, J., Generalized secret sharing and monotone functions, *Proceedings of Crypto'88, LNCS*, Vol. 403, pp. 27–35, Springer-Verlag, Berlin, 1989.
8. Bennett, C.H., Quantum cryptography using any two nonorthogonal states, *Physical Review Letters,* 68, 3121–3124, 1992.
9. Bennett, C.H., Bessette, F., Brassard, G., Salvail, L., and Smolin, J., Experimental quantum cryptography, *Journal of Cryptology,* 5, 3–28, 1992.
10. Bennett, C.H. and Brassard, G., Quantum cryptography: Public-key distribution and coin tossing, *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing,* IEEE, pp. 175–179, New York, 1984.
11. Bennett, C.H., Brassard, G., and Ekert, A.K., Quantum cryptography, *Scientific American,* pp. 50–57, Oct. 1992.

12. Bennett, C.H., Brassard, G., and Robert, J.-M., Privacy amplification by public discussion, *SIAM Journal on Computing,* 17(2), 210–229, 1988.
13. Bertilson, M. and Ingemarsson, I., A construction of practical secret sharing schemes using linear block codes, *Proceedings of AUSCRYPT'92, LNCS*, Vol. 718, pp. 67–79, Springer-Verlag, Berlin, 1993.
14. Beth, T., Jungnickel, D., and Lenz, H., *Design Theory, 2nd edition,* Cambridge University Press, Cambridge, U.K., 1999.
15. Beth, T., Multifeature security through homomorphic encryption, *Proceedings of Asiacrypt'94, LNCS*, Vol. 917, pp. 3–17, Springer-Verlag, Berlin, 1993.
16. Beutelspacher, A., Enciphered geometry: Some applications of geometry to cryptography, *Discrete Applied Mathematics,* 37, 59–68, 1988.
17. Blakley, G.R., Safeguarding cryptographic keys, *Proceedings of N.C.C., AFIPS Conference Proceedings*, Vol. 48, pp. 313–317, Montvale, NJ, 1979.
18. Blakley, B., Blakley, G.R., Chan, A.H., and Massey, J.L., Threshold schemes with disenrollment, *Proceedings of Crypto'92, LNCS*, Vol. 740, pp. 546–554, Springer-Verlag, Berlin, 1992.
19. Blakley, G.R. and Meadows, C., Security of ramp schemes, *Proceedings of CRYPTO'84, LNCS*, Vol. 196, pp. 242–268, Springer-Verlag, Berlin, 1985.
20. Blundo, C., De Santis, A., Stinson, D.R., and Vaccaro, V., Graph decompositions and secret sharing schemes, *Journal of Cryptology,* 8(1), 39–64, 1995.
21. Boyd, C., Digital Multisignatures, In *Cryptography and Coding,* Beker, H. and Piper, F., Eds., pp. 241–246. Clarendon Press, Gloucestershire, GL, 1989.
22. Brassard, G., *Modern Cryptology: A Tutorial,* Springer, Berlin, 1988.
23. Brickell, E.F., Some ideal secret sharing schemes, *Journal of Combinatorial Mathematics and Combinatorial Computing,* 6, 105–113, 1989.
24. Brickell, E.F. and Davenport, D.M., On the classification of ideal secret sharing schemes, *Journal of Cryptology,* 4, 123–134, 1991.
25. Brickell, E.F. and Stinson, D.R., The detection of cheaters in threshold schemes, *Proceedings of Crypto'88, LNCS*, Vol. 403, pp. 564–577, Springer-Verlag, Berlin, 1990.
26. Brickell, E.F. and Stinson, D.R., Some improved bounds on the information rate of perfect secret sharing schemes, *Journal of Cryptology,* 5, 153–166, 1992.
27. Brown, L., Kwan, M., Pieprzyk, J., and Seberry, J., Improving resistance to differential cryptanalysis and the redesign of LOKI, in *Advances in Cryptology—Proceedings of ASIACRYPT '91,* Imai, R.R.H. and Matsumoto, T., Eds., Vol. 739 of *Lecture Notes in Computer Science*, pp. 36–50, Springer-Verlag, Berlin, 1993.
28. Capocelli, R., DeSantis, A., Gargano, L., and Vaccaro, V., On the size of shares for secret sharing schemes, *Proceedings of CRYPTO '91, LNCS*, Vol. 576, pp. 101–113, Springer-Verlag, Berlin, 1992.
29. Cerecedo, M., Matsumoto, T., and Imai, H., Efficient and secure multiparty generation of digital signatures based on discrete logarithms, *IEICE Transactions on Fundamentals* of *Electronics, Communications, and Computer Science*, E76-A(4), 531–545, Apr. 1993.
30. Chang, C.-C. and Liou, F.-Y., A digital multisignature scheme based upon the digital signature scheme of a modified ElGamal public key cryptosystem, *Journal of Information Science and Engineering,* 10, 423–432, 1994.
31. Charnes, C., Pieprzyk, J., and Safavi-Naini, R., Families of threshold schemes, *Proceedings of 1994 IEEE International Symposium on Information Theory,* Trondheim, Norway, 1994.
32. Charnes, C., Pieprzyk, J., and Safavi-Naini, R., Conditionally secure secret sharing schemes with disenrollment capability, 2*nd ACM Conference on Computer and Communications Security,* Nov. 2–4, Fairfax, VA, pp. 89–95, ACM 1994.
33. Charnes, C. and Pieprzyk, J., Disenrollment capability of conditionally secure secret sharing schemes, *Proceedings of International Symposium on Information Theory and Its Applications (ISITA'94),* Nov. 20–25, 1994, pp. 225–227, Sydney, Australia, IEA, NCP 94/9 1994.

34. Charnes, C. and Pieprzyk, J., Cumulative arrays and generalised Shamir secret sharing schemes, 17*th Australasian Computer Science Conference, Australian Computer Science Communications*, 16(1), 519–528, 1994.
35. Charnes, C. and Pieprzyk, J., Generalised cumulative arrays and their application to secret sharing schemes, 18*th Australasian Computer Science Conference, Australian Computer Science Communications*, 17(1), 61–65, 1995.
36. Chaudhry, G. and Seberry, J., Secret sharing schemes based on Room squares, *Combinatorics, Complexity and Logic, DMTCS'96*, pp. 158–167, Springer-Verlag, Berlin, 1996.
37. Chor, B., Goldwasser, S., Micali, S., and Awerbuch, B., Verifiable secret sharing and achieving simultaneity in the presence of faults, *Proceedings of 26th Annual IEEE Symposium on Foundations of Computer Science*, pp. 383–395, IEEE, Portland 1985.
38. Cooper, J., Donovan, D., and Seberry, J., Secret sharing schemes arising from Latin squares, *Bulletin of the ICA,* 12, 33–43, 1994.
39. Damgard, I. and Thorbek, R., Linear integer secret sharing and distributed exponentiation, http://eprint.iacr.org/2006/044.
40. Desmedt, Y., Society and group oriented cryptography: A new concept, In *Advances in Cryptology—Proceedings of CRYPTO '87,* Pomerance, C., Ed., LNCS, Vol. 293, pp. 120–127, Springer-Verlag, Berlin, 1988.
41. Desmedt, Y. and Frankel, Y., Threshold cryptosystems, in *Advances in Cryptology—Proceedings of CRYPTO '89,* Brassard, G., Ed., Vol. 435 of *Lecture Notes in Computer Science*, pp. 307–315, Springer-Verlag, Berlin, 1990.
42. Desmedt, Y. and Frankel, Y., Shared generation of authenticators and signatures, in *Advances in Cryptology—Proceedings of CRYPTO'91,* Feigenbaum, J., Ed., LNCS, Vol. 576, pp. 457–469, Springer-Verlag, Berlin, 1992.
43. Deutsch, D., Quantum theory, the Church-Turing principle and the universal quantum computer, *Proceedings of the Royal Society of London, Series A,* 400, 96–117, 1985.
44. Deutsch, D., Quantum computational networks, *Proceedings of the Royal Society of London, Series A,* 425, 73–90, 1989.
45. Diffie, W. and Hellman, M., New directions in cryptography, *IEEE Transactions on Information Theory*, IT-22, 644–654, Nov. 1976.
46. DiVincenzo, D.P., Quantum computation, *Science,* 270, 255–261, 1995.
47. ElGamal, T., A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Transactions on Information Theory*, IT-31, 469–472, Jul. 1985.
48. Ekert, A.K., Quantum cryptography based on Bell's theorem, *Physical Review Letters,* 67, 661–663, 1991.
49. Ekert, A.K., Rarity, J.G., Tapster, P.R., and Palma, G.M., Practical quantum cryptography based on two-photon interferometry, *Physical Review Letters,* 69, 1293–1295, 1992.
50. Ekert, A.K., and Josza, R., Shor's algorithm for factoring numbers, *Review of Modern Physics,* 68(30), 733–753, 1996.
51. Feldman, J., Malkin, J., Servedio, R.A., and Stein, C., On the capacity of secure network coding, *Proceedings of the 42nd Annual Allerton Conference on Communication, Control and Computing*, Urbana, IL, 2004.
52. Gallagher, R.G., *Information Theory and Reliable Communications,* John Wiley & Sons, New York, 1968.
53. Ghodosi, H., Pieprzyk, J., and Safavi-Naini, R., Dynamic threshold cryptosystems, *Proceedings of PRAGOCRYPT'96,* CTU Publishing House, Prague, Part 1, pp. 370–379, 1996.
54. Gisin, N., Ribordy, G., Titel, W., and Zbinden, H., Quantum cryptography, *Reviews of Modern Physics,* 74(1), 145–195, 2002.
55. Harn, L., Group-oriented (t, n) threshold digital signature scheme and digital multisignature, *IEEE Proceedings—Computers and Digital Techniques*, 141(5), 307–313, Sep. 1994.

56. Harn, L. and Yang, S., Group-oriented undeniable signature schemes without the assistance of a mutually trusted party, In *Advances in Cryptology—Proceedings of AUSCRYPT '92,* Seberry, J. and Zheng, Y., Eds., *LNCS*, Vol. 718, pp. 133–142, Springer-Verlag, Berlin, 1993.
57. Hwang, T., Cryptosystem for group oriented cryptography, *Proceedings of Eurocrypt'90, LNCS*, Vol. 473, pp. 353–360, Springer-Verlag, Berlin, 1990.
58. Hwang, S.-J. and Chang, C.-C., A dynamic secret sharing scheme with cheater detection, *Proceedings of ACISP'96, LNCS*, Vol. 1172, NSW, Australia, pp. 48–55, 1996.
59. Imai, H., Mueller-Quade, J., Tuyls, P., Nascimento, A., Tuyls, P., and Winter, A., An information theoretical model for quantum secret sharing schemes, *Quantum Information and Computation,* 5(1), 69–80, 2005.
60. Itakura, K. and Nakamura, K., A public-key cryptosystem suitable for digital multisignature, *NEC Research & Development*, 71, Oct. 1983.
61. Ito, M., Saito, A., and Nishizeki, T., Secret sharing scheme realising general access structure, *Proceedings of Globecom'87,* 99–102, Tokyo, 1987.
62. Jackson, W.-A. and Martin, K.M., Cumulative arrays and geometric secret sharing schemes, *Proceedings of Auscrypt'92, LNCS*, Vol. 718, pp. 49–55, Springer-Verlag, Berlin, 1993.
63. Karnin, E.D., Greene, J.W., and Hellman, M.E., On secret sharing systems, *IEEE Transactions on Information Theory*, IT-29(1) 35–41, 1983.
64. Krawczyk, H., Secret sharing made short, *Proceedings of Crypto'93, LNCS*, Vol. 773, pp. 136–146, Springer-Verlag, Berlin, 1994.
65. Laih, C.-S. and Yen, S.-M., Multi-signature for specified group of verifiers, *Journal of Information Science and Engineering,* 12(1), 143–152, Mar. 1996.
66. Langford, S.K., Threshold DSS signatures without a trusted party, *Proceedings of Crypto'95, LNCS*, Vol. 963, pp. 397–409, Springer-Verlag, Berlin, 1996.
67. Lanyon, B.P. et al., Experimental demonstration of Shor's algorithm with quantum entanglement, *Physical Review Letters,* 99, 250505, 2007. arXiv:quant-ph/07051398.
68. Li, C.-M., Hwang, T., and Lee, N.-Y., Threshold-multisignature schemes where suspected forgery implies traceability of adversarial shareholders, In *Advances in Cryptology—Proceedings of EUROCRYPT '94,* De Santis, A., Ed., *LNCS* Vol. 950, pp. 194–204, Springer-Verlag, Berlin, 1995.
69. Lin, H.-Y. and Harn, L., A generalized secret sharing scheme with cheater detection. *Proceedings of Asiacrypt'91, LNCS*, Vol. 739, pp. 149–158, Springer-Verlag, Berlin, 1993.
70. Lo, H. and Lütkenhaus, N. Quantum cryptography: From theory to practice, arXiv:quant-ph/0702202. A generalized secret sharing scheme with cheater detection.
71. Massey, J.L., Minimal codewords and secret sharing, *Proceedings of 6th Joint Swedish-Russian Workshop in Information Theory,* Molle, Sweden, pp. 276–279, 1993.
72. Massey, J.L., Provable security–myth or reality?, *ISPEC*, Hangzhou, China, 2006.
73. Martin, K. and Jackson, W.-A., Perfect secret sharing schemes on five participants, *Designs Codes and Cryptography,* 9, 267–286, 1996.
74. Maurer, W., Helwig, W., and Silberhorn, C., Recent developments in quantum key distribution: Theory and practice, *Annale of Physics (Leipzig),* No. 2–3, 2008. arXiv:quant-ph/0712.0517.
75. McEliece, R.J. and Sarwate, D.V., On sharing secrets and Reed-Solomon codes, *Communications of the ACM,* 24(9), 683–584, 1981.
76. MacWilliams and Sloane, N.J.A., *Theory of Codes,* North Holland, Amsterdam, the Netherlands 1977.
77. Muller, A., Breguet, J., and Gisin, N., Experimental demonstration of quantum cryptography using polarised photons in optical fibre over more than 1 km, *Europhysics Letters,* 23, 383–388, 1993.
78. National Bureau of Standards, Federal Information Processing Standard (FIPS), *Data Encryption Standard*, 46 edn., U.S. Department of Commerce, Washington, DC, Jan. 1977.
79. Ogata, W. and Kurosawa, K., Lower bound on the size of shares of nonperfect secret sharing schemes, *Proceedings of Asiacrypt'94, LNCS*, Vol. 917, pp. 33–41, Springer-Verlag, Berlin, 1995.

80. Ogata, W., Kurosawa, K., and Tsujii, S., Nonperfect secret sharing schemes, *Proceedings of Auscrypt'92, LNCS*, Vol. 718, pp. 56–66, Springer-Verlag, Berlin, 1993.
81. Ogata, W. and Kurosawa, K., MDS secret sharing scheme secure against cheaters, *IEEE Transactions on Information Theory,* 46(3), 1078–1081, 2000.
82. Ohta, K. and Okamoto, T., A digital multisignature scheme based on the Fiat-Shamir scheme, In *Advances in Cryptology—Proceedings of ASIACRYPT '91,* Rivest, R.L., Imai, H., and Matsumoto, T., Eds., Vol. 739 of *Lecture Notes in Computer Science,* pp. 139–148, Springer-Verlag, Berlin, 1993.
83. Park, C. and Kurosawa, K., New ElGamal type threshold digital signature scheme, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E79(1), 86–93, Jan. 1996.
84. Renvall, A. and Ding, C., The access structure of some secret sharing schemes, *Proceedings of ACISP'96, LNCS*, Vol. 1172, pp. 67–86, Springer-Verlag, Berlin, 1996.
85. Rivest, R., Shamir, A., and Adleman, L., A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM,* Vol. 21, pp. 120–126, Feb. 1978.
86. Rivest, R., Shamir, A., and Adleman, L., On digital signatures and public-key cryptosystems, MIT Laboratory for Computer Science, Technical Report, MIT/LCS/TR-212, Jan. 1979.
87. Scarani, V., Acin, A., Ribordy, V., and Gisin, V., *Physical Review Letters,* 92, 057901, 2004.
88. Schellenberg, P.J. and Stinson, D.R., Threshold schemes from combinatorial designs, *Journal of Combinatorial Mathematics and Combinatorial Computing,* 5, 143–160, 1989.
89. Schmidt-Kaler, F. et al., Realization of the Cirac-Zoller controlled-NOT quantum gate, *Nature,* 422, 408–411, 2003.
90. Schneier, B., *Applied Cryptography: Protocols, Algorithms, and Source Code in C,* John Wiley & Sons, New York, 1994.
91. Pieprzyk, J., Hardjono, T., and Seberry, J., *Fundamentals of Computer Security,* Springer-Verlag, Berlin, 2003.
92. Shamir, A., How to share a secret, *Communications of the ACM,* 22(11), 612–613, 1979.
93. Shimizu, A. and Miyaguchi, S., Fast data encipherment algorithm FEAL, *Advances in Cryptology—Proceedings of EUROCRYPT '87,* Chaum, D. and Price, W., Eds., Vol. 304 of *Lecture Notes in Computer Science*, pp. 267–278, Springer-Verlag, Berlin, 1987.
94. Shor, P.W., Algorithms for quantum computation: Discrete log and factoring, *Proceedings of the 35th Annual Symposium on the Foundations of Computer Science,* pp. 124–134, Goldwasser, S., Ed., IEEE Computer Society Press, Los Alamitos, CA, 1994.
95. Simmons, G.J., How to (really) share a secret, *Proceedings of Crypto'88, LNCS*, Vol. 403, pp. 390–448, Springer-Verlag, Berlin, 1989.
96. Simmons, G.J., An introduction to shared secret and/or shared control schemes and their application, In *Contemporary Cryptology—The Science of Information Integrity,* pp. 441–497, Simmons, G.J., Ed., IEEE Press, New York, 1992.
97. Simmons, G.J., Jackson, W.-A., and Martin, K., The geometry of shared secret schemes, *Bulletin of the ICA,* 1, 71–88, 1991.
98. De Soete, M., Quisquater, J.-J., and Vedder, K., A signature with shared verification scheme, In Brassard, J., Ed., *Advances in Cryptology—Proceedings of CRYPTO '89,* Vol. 435 of *Lecture Notes in Computer Science,* pp. 253–262. Springer-Verlag, Berlin, 1990.
99. NSBS. Processing Information Systems. Cryptographic Protection. Cryptographic Algorithm, GOST 28147-89, C1–C26, National Soviet Bureau of Standards, 1989.
100. Street, A.P., Defining sets for t-designs and critical sets for Latin squares, *New Zealand Journal of Mathematics,* 21, 133–144, 1992.
101. Scarani, V. at al., A framework for practical quantum cryptography, arXiv:quant-ph/0802.4155.
102. Stinson, D.R., An explication of secret sharing schemes, *Designs, Codes and Cryptography,* 2, 357–390, 1992.
103. Stinson, D.R., Bibliography on secret sharing schemes, http://www.cacr.math.uwaterloo.ca/dstinson/pubs.html.

104. Stinson, D.R., *Cryptography Theory and Practice,* 3rd Edition, CRC Press, Boca Raton, FL, 2005.
105. Stinson, D.R. and Vanstone, S.A., A combinatorial approach to threshold schemes, *SIAM Journal of Discrete Mathematics,* 1, 230–236, 1988.
106. Tapster, P.R., Rarity, J.G., and Owens, P.C.M., Violation of Bell's inequality over 4 km of optical fibre, *Physical Review Letters,* 73, 1923–1926, 1994.
107. Tompa, M. and Woll, H., How to share a secret with cheaters, *Journal of Cryptology,* 1, 133–138, 1988.
108. Townsend, P.D., Rarity, J.G., and Tapster, P.R., Enhanced single photon fringe visibility in a 10 km-long prototype quantum cryptography channel, *Electronic Letters,* 29, 1291–1293, 1993.
109. Vandersypen, L.K.M. et al., Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance, *Nature,* 414, 883-887, 2001.
110. Wiesner, S., Conjugate coding, *SIGACT News,* 15, 78–88, 1983. (Original manuscript written circa 1970.)