1 Applications of Designs to Cryptography

K. Gopalakrishnan and D. R. Stinson

1.1 Secrecy Codes

One fundamental goal of cryptography is to allow two people, who we will call Alice and Bob, to communicate over an insecure channel in such a way that an observer, Oscar, cannot understand what is being said. Alice encrypts the plaintext message, using a predetermined key, and sends the resulting ciphertext over the channel. Even if Oscar intercepts the ciphertext, he cannot determine what the plaintext was; but Bob, who knows the key, can decrypt the ciphertext and obtain the plaintext. This can be described more formally as follows.

A *cryptosystem* or *secrecy code* is a five-tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where the following conditions are satisfied:

1. $\mathcal{P}$ is a finite set of possible *plaintexts*.
2. $\mathcal{C}$ is a finite set of possible *ciphertexts*
3. $\mathcal{K}$, the *keyspace*, is a finite set of possible *keys*.
4. For each $K \in \mathcal{K}$, there is an injective *encryption rule* $e_K \in \mathcal{E}$ and a corresponding *decryption rule* $d_K \in \mathcal{D}$. Each $e_K : \mathcal{P} \to \mathcal{C}$ and $d_K : \mathcal{C} \to \mathcal{P}$ are such that $d_K(e_K(x)) = x$ for every plaintext $x \in \mathcal{P}$.

Alice and Bob jointly choose a secret random key $K \in \mathcal{K}$. At a later time, when Alice wants to send $x \in \mathcal{P}$ to Bob, she encrypts x using the encryption rule e_K. That is, she computes $y = e_K(x)$, and sends y to Bob over the channel. When Bob receives y, he decrypts it using the decryption function d_K, obtaining x.

Informally, *perfect secrecy* means that observation of a ciphertext gives no information about the corresponding plaintext. This idea can be stated more precisely using probability distributions. Suppose there is are probability distributions $p_{\mathcal{P}}$ on $\mathcal{P}$, and $p_{\mathcal{K}}$ on $\mathcal{K}$. Then a probability distribution $p_{\mathcal{C}}$ is induced on $\mathcal{C}$. A cryptosystem is said to provide *perfect secrecy* provided that

$$p_{\mathcal{P}}(x|y) = p_{\mathcal{P}}(x) \text{ for every } x \in \mathcal{P} \text{ and } y \in \mathcal{C}.$$

Shannon proved the following fundamental result in 1949. This is probably the first result linking design theory and cryptography.

1.1 Theorem In any cryptosystem that provides perfect secrecy, $|\mathcal{K}| \geq |\mathcal{P}|$. Further, equality occurs if and only if the encryption matrix is a latin square of order $|\mathcal{P}|$ and the encryption rules are used with equal probability.

1.2 Example The well-known *Vernam One-time Pad* is defined as follows. $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbf{Z}_2)^n$ for some integer $n \geq 1$. For any $x = (x_1, \ldots, x_n) \in (\mathbf{Z}_2)^n$ and $K = (K_1, \ldots, K_n) \in (\mathbf{Z}_2)^n$, we define

$$e_K(x) = (x_1 + K_1 \bmod 2, \ldots, x_n + K_n \bmod 2).$$

It is easily seen that the encryption matrix is a latin square of order 2^n, and thus perfect secrecy is achieved.

The above discussion assumes that each key is used for only one encryption. The theory can be extended in a natural way, as follows. Suppose each key is used to encrypt up to t plaintexts, where $t \geq 1$ is an integer. *Perfect t-fold secrecy* means that observation of a set of at most t ciphertexts gives no information about the corresponding set of t plaintexts. As before, we use probability distributions to define the notion precisely. For any $s \leq t$, suppose there is a probability distribution $p_{\binom{\mathcal{P}}{s}}$ on the set of s-subsets of $\mathcal{P}$. Also, suppose there is a probability distribution $p_{\mathcal{K}}$ on $\mathcal{K}$. Then, for each $s \leq t$, a probability distribution $p_{\binom{\mathcal{C}}{s}}$ is induced on the set of s-subsets of $\mathcal{C}$. Let $P(s)$ denote the following property:

$$p_{\binom{\mathcal{P}}{s}}(X|Y) = p_{\binom{\mathcal{P}}{s}}(X) \text{ for every } X \subseteq \mathcal{P} \text{ and } Y \subseteq \mathcal{C} \text{ such that } |X| = |Y| = s.$$

A cryptosystem is said to provide *perfect t-fold secrecy* provided that $P(s)$ is satisfied for $1 \leq s \leq t$.

1.3 Theorem Suppose a perpendicular array $PA_\lambda(t, k, v)$ with $k \geq 2t - 1$ exists. Then a cryptosystem (with k plaintexts, v ciphertexts and $\lambda\binom{v}{t}$ encryption rules) that provides perfect t-fold secrecy exists.

Here is a sketch of a proof of this result. The *encryption matrix* of a cryptosystem is defined to be the $|\mathcal{K}| \times |\mathcal{P}|$ matrix in which the entry in row K ($K \in \mathcal{K}$) and column x ($x \in \mathcal{P}$) is $e_K(x)$. If we use a $PA_\lambda(t, k, v)$ as the encryption matrix of a cryptosystem, and choose each of the $\lambda\binom{v}{t}$ encryption rules with equal probability, then it is not difficult to see that property $P(t)$ is satisfied. However, we need property $P(s)$ to be satisfied for all $1 \leq s \leq t$. In other words, we desire that our $PA_\lambda(t, k, v)$ is also a $PA_{\lambda(s)}(s, k, v)$ for $1 \leq s \leq t$. Such a perpendicular array is called *inductive*. Any $PA_\lambda(t, k, v)$ is inductive provided $k \geq 2t - 1$. Hence the result follows.

The following theorem gives a partial converse.

1.4 Theorem

Suppose a cryptosystem provides perfect t-fold secrecy, and $|\mathcal{P}| \geq 2t - 1$. Then $|\mathcal{K}| \geq \binom{\mathcal{P}}{t}$, and equality occurs if and only if the encryption matrix is a perpendicular array $PA(t, |\mathcal{P}|, |\mathcal{P}|)$ and the encryption rules are used with equal probability.

1.2 Authentication Codes

Authentication codes were invented in 1974 by Gilbert, MacWilliams and Sloane and have been extensively studied in recent years. The general theory of unconditional authentication was developed by Simmons. Recent research in authentication codes has involved considerable use of various combinatorial designs. In this section, we will consider three different types of authentication codes (viz., authentication codes without secrecy, general authentication codes and authentication codes with secrecy) and their relation to combinatorial designs.

1.2.1 Authentication without secrecy

As above, we have the three participants Alice, Bob and Oscar. Suppose that Oscar has the ability to introduce his own messages into the channel and/or to modify existing messages. The purpose of an authentication code is to protect the *integrity* of the message. That is, when Bob receives a message from Alice, he wants to be sure that it was really sent by Alice and was not tampered with by Oscar.

We consider two types of attacks. When Oscar places a (new) message m' into the channel, this is called *impersonation*. When Oscar sees a message m and changes it to a message $m' \neq m$, this is called *substitution*.

Here is our first definition of an authentication code.

An *authentication code without secrecy* is a four-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{K}, \mathcal{E})$, where the following conditions are satisfied:

1. $\mathcal{S}$ is a finite set of plaintexts called *source states*.
2. $\mathcal{A}$ is a finite set of *authenticators*.
3. $\mathcal{K}$ is a finite set of *keys*.
4. For each $K \in \mathcal{K}$, there is an *authentication rule* $e_K \in \mathcal{E}$. Each $e_K : \mathcal{S} \to \mathcal{A}$.

As before, Alice and Bob jointly choose a secret key $K \in \mathcal{K}$. At a later time, when Alice wants to communicate a source state $s \in \mathcal{S}$ to Bob, she uses the authentication rule e_K to produce the authenticator $a = e_K(s)$. The *message* $m = (s, a)$ is then sent over the channel. When Bob receives m, he verifies that $a = e_K(s)$ to authenticate the source state s.

When Oscar performs impersonation or substitution, his goal is to have his bogus message $m' = (s', a')$ accepted as authentic by Bob, thus misleading Bob as to the state of the source. That is, if e is the authentication rule being used (which is not known to Oscar), then he is hoping that $a' = e(s')$.

We assume that there is some probability distribution on $\mathcal{S}$, which is known to all the participants. Given this probability distribution, Alice and Bob will choose a probability distribution for $\mathcal{E}$, called an *authentication strategy*. Once Alice and Bob have chosen their authentication strategy, it is possible to compute, for $i = 0, 1$, a *deception probability* denoted by P_{d_i}, which is the probability that Oscar can deceive Bob by impersonation and substitution, respectively. In computing the deception probabilities, it is assumed that Oscar is using an optimal strategy.

It is not difficult to show that $P_{d_0} \geq 1/|\mathcal{A}|$ and $P_{d_1} \geq 1/|\mathcal{A}|$ as well. We are not only interested in minimizing the deception probabilities but also in minimizing the number of authentication rules in the code. This is because the key, K, must be exchanged secretly between Alice and Bob, and thus we are minimizing the amount of information that must be kept secret.

We represent an authentication code by the $|\mathcal{E}| \times |\mathcal{S}|$ *authentication matrix*, in which the rows are indexed by authentication rules, the columns are indexed by source states, and the entry in row e and column s is $e(s)$. Authentication codes (without secrecy) with minimum possible deception probabilities and the minimum number of encoding rules have been shown to be equivalent to orthogonal arrays.

1.5 Theorem Suppose we have an authentication code without secrecy for k source states and having l authenticators, in which $P_{d_0} = P_{d_1} = 1/l$. Then

1. $|\mathcal{E}| \geq l^2$, and equality occurs if and only if the authentication matrix is an orthogonal array $OA(2, k, l)$ (with $\lambda = 1$) and the authentication rules are used with equal probability;
2. $|\mathcal{E}| \geq k(l-1) + 1$, and equality occurs if and only if the authentication matrix is an orthogonal array $OA_\lambda(2, k, l)$ where
$$\lambda = \frac{k(l-1)+1}{l^2},$$
and the authentication rules are used with equal probability.

Here is an extension of our impersonation/substitution model. Suppose the same encoding rule is used for the transmission of i messages, which Oscar observes in the channel. Then, suppose that Oscar inserts a new message of his own choosing, hoping to have it accepted by Bob as authentic. This is called *spoofing* of *order i*. The deception probability P_{d_i} is the probability that Oscar will succeed in deceiving Bob with an order i spoofing attack. An authentication code is *t-fold secure against* *spoofing* if $P_{d_i} = 1/l$ for $0 \le i \le t$. Such codes are related to orthogonal arrays of strength t as indicated by the following theorem.

1.6 Theorem

Suppose we have an authentication code without secrecy that is t-fold secure against spoofing. Then $|\mathcal{E}| \ge |\mathcal{A}|^t$, and equality occurs if and only if the authentication matrix is an orthogonal array $OA(t, |\mathcal{S}|, |\mathcal{E}|)$ and the authentication rules are used with equal probability.

This is actually an easy generalization of the first part of the previous theorem. The generalization of the second part to the case of t-fold secure authentication codes is still an open problem.

1.2.2 General authentication codes

In the case of general authentication codes, we have an arbitrary message space $\mathcal{M}$; i.e., a message does not consist of a source state with an appended authenticator. In this scenario, it is not a requirement that the message reveals the source state. The following bounds on deception probabilites hold for general codes:

$$P_{d_0} \ge \frac{|\mathcal{S}|}{|\mathcal{M}|}$$

and

$$P_{d_1} \ge \frac{|\mathcal{S}| - 1}{|\mathcal{M}| - 1}.$$

Codes satisfying these bounds with equality and having the minimum number of authentication rules have been characterized in terms of Steiner 2-designs $S(2, k, v)$:

1.7 Theorem

Suppose $P_{d_0} = |\mathcal{S}|/|\mathcal{M}|$ and $P_{d_1} = (|\mathcal{S}| - 1)/(|\mathcal{M}| - 1)$. Then

$$|\mathcal{E}| \ge \frac{|\mathcal{M}|(|\mathcal{M}| - 1)}{|\mathcal{S}|(|\mathcal{S}| - 1)},$$

and equality occurs if and only if the authentication matrix is an $S(2, |\mathcal{S}|, |\mathcal{M}|)$, the authentication rules are used equiprobably, and the source states occur equiprobably.

t-fold secure general authentication codes have been studied as well. In the general case, the deception probabilites are bounded below according to the formula

$$P_{d_i} \ge \frac{|\mathcal{S}| - i}{|\mathcal{M}| - i}.$$

A general authentication code is defined to be *t-fold secure against spoofing* if

$$P_{d_i} = \frac{|\mathcal{S}| - i}{|\mathcal{M}| - i}.$$

for $0 \le i \le t$.

1.8 Theorem

Suppose a general authentication code is t-fold secure against spoofing. Then

$$|\mathcal{E}| \ge \frac{\binom{|\mathcal{M}|}{t+1}}{\binom{|\mathcal{S}|}{t+1}},$$

and equality occurs only if the authentication matrix is an $S(t+1, |\mathcal{S}|, |\mathcal{M}|)$. Conversely, if there exists an $S(t+1, k, v)$, then there exists an authentication code for k equiprobable source states, having v possible messages and $\binom{v}{t+1}/\binom{k}{t+1}$ equiprobable authentication rules, that is t-fold secure against spoofing.

1.2.3 Authentication with secrecy

We will now consider authentication codes that are t-fold secure against spoofing and simultaneously provide perfect t'-fold secrecy. We will denote such codes as (t, t') codes.

First we will consider the simple case of $t = t' = 1$. Such codes can be constructed from Steiner 2-designs that satisfy an additional numerical condition.

1.9 Theorem

Suppose there is an $S(2, k, v)$, where $v - 1 \equiv 0 \bmod k(k-1)$. Then there is a $(1, 1)$-code for k equiprobable source states, having v messages and $v(v-1)/(k(k-1))$ equiprobable encoding rules.

The authentication capability of the code is obvious as it is analogous to the situation of general authentication codes. To provide for perfect secrecy, we need to order every block of the $S(2, k, v)$ in such a way such that every point occurs in each possible position in exactly $(v-1)/(k(k-1))$ blocks. Clearly, a necessary condition for this to be possible is $v - 1 \equiv 0 \bmod k(k-1)$, since every point occurs in exactly $(v-1)/(k-1)$ blocks. That this condition is also sufficient can be demonstrated by finding suitable systems of distinct representatives.

Finally, we will consider $(t-1, t)$ codes. We saw in the last section that (inductive) perpendicular arrays $PA_\lambda(t, k, v)$ yield codes having t-fold secrecy. If we employ a PA that enjoys an extra property, the code constructed from a $PA_\lambda(t, k, v)$ will also be $(t-1)$-fold secure against spoofing. This motivates the following definition. An *authentication perpendicular array* $APA_\lambda(t, k, v)$ is a $PA_\lambda(t, k, v)$ such that, for any $s \le t - 1$ and for any $s + 1$ distinct symbols x_i $(1 \le i \le s+1)$, we have that among the rows of the PA that contain all the symbols x_i $(1 \le i \le s+1)$, the s symbols x_i $(1 \le i \le s)$ occur in all possible subsets of s columns equally often. This property may seem complicated, but is precisely what is needed for a $PA_\lambda(t, k, v)$ to be $(t-1)$-fold secure against spoofing.

1.10 Example

As an example to illustrate the definition, we present an $APA(2, 3, 11)$. Construct a 55×3 array A by developing the following five rows modulo 11:

$$\begin{array}{ccc} 0 & 1 & 2 \\ 0 & 9 & 7 \\ 0 & 3 & 6 \\ 0 & 4 & 8 \\ 0 & 5 & 10 \end{array}$$

Any unordered pair $\{x, y\}$ will occur in three rows of A. Within these three rows, x will occur once in each of the three columns, as will y.

1.11 Theorem

If there exists an $APA_\lambda(t, k, v)$, then there exists an authentication code that is $(t-1)$-fold secure against spoofing and provides t-fold secrecy for k equiprobable source states, having v messages and $\lambda\binom{v}{t}$ encoding rules. The code is optimal if and only if $\lambda = 1$.

1.3 Threshold schemes

Suppose that a bank has a vault that must be opened every day. The bank employs three senior tellers, but they do not want to trust any individual with the combination. Hence, they would like a system whereby any two of the three senior tellers can gain access to the vault. This problem can be solved by means of threshold schemes.

Threshold schemes, which are a special case of secret sharing schemes, were first described by Shamir and Blakley in 1979. Blakley's solution uses finite geometries, while Shamir's scheme is based on polynomial interpolation (actually orthogonal arrays in disguise).

A *perfect* (t, w)*-threshold scheme* is a method of sharing a secret key K among a finite set $\mathcal{P}$ of w participants, in such a way that any t participants can compute the value of K, but no group of $t-1$ (or fewer) participants can compute any information about the value of k (in an information-theoretic sense) from the shares they hold.

The value of K is chosen by a special participant called the *dealer*. The dealer is denoted by D and we assume $D \notin \mathcal{P}$. When D wants to share the key K among the participants in $\mathcal{P}$, he gives each participant some partial information called a *share*. The shares should be distributed secretly, so no participant knows the share given to another participant.

At a later time, a subset of participants $B \subseteq \mathcal{P}$ will pool their shares in an attempt to compute the secret key K. If $|B| \geq t$, then they should be able to compute the value of K as a function of the shares they collectively hold; if $|B| < t$, then they should not be able to compute K. In the bank vault example described above, we desire a $(2, 3)$-threshold scheme.

Each share is chosen from a specified *share set* $\mathcal{S}$. It is well known that in any perfect threshold scheme $|\mathcal{K}| \leq |\mathcal{S}|$. A schemes in which equality is attained is called an *ideal scheme*.

The first result we want to mention in this section is a combinatorial characterization of ideal threshold schemes.

1.12 Theorem

An ideal (t, w)-threshold scheme with $|\mathcal{K}| = v$ is equivalent to an $OA(t, w + 1, v)$.

It is easy to obtain a threshold scheme from an orthogonal array. We associate the first w columns of the array with the w participants, and the last column with the key. If the dealer wants to share key K, then he chooses a random row of the array such that K occurs in the last column, and gives out the remaining w elements in that row as the shares. Now, any t shares uniquely determine a row of the array, and the key is revealed as the element in the last column. But $t-1$ shares leave the key completely undetermined, since for any hypothetical value of the key, say K, there is a unique row containing the $t-1$ given shares and having K in the last column.

We now turn our attention to a special type of threshold scheme which is known as "anonymous threshold scheme."

A perfect threshold scheme is called *anonymous* if the following two properties are satisfied:

1. The w participants receive w distinct shares.
2. The key can be computed solely as a function of t shares, without the knowledge of which participant holds which share. Thus, the key computation can be performed by a black box that is given t shares and does not know the identities of the participants holding those shares.

1.13 Theorem

In an anonymous (t, w)-threshold scheme, $|\mathcal{S}| \geq |\mathcal{K}|(w - t + 1) + t - 1$. Further, equality occurs if and only if there exists an $S(t, w, v)$ that can be partitioned into $S(t - 1, w, v)$.

1.14 Example

As an illustration of the above theorem, let us see how a large set of $S(2, 3, 9)$ can be used to construct an anonymous $(3, 3)$ threshold scheme with seven possible keys. The large set consists of seven disjoint $S(2, 3, 9)$. Denote them by $\mathcal{B}_i$ $(0 \leq i \leq 6)$. Now, the seven keys correspond to the seven designs in the large set; the shares are the points in X. If the dealer's secret is K, then he chooses a random block B from $\mathcal{B}_K$ and gives the three points in B to the three participants as their shares. Now, three shares uniquely determine the block B, which occurs in exactly one of the seven designs (since we have a large set). Hence three participants can compute K. But from two shares, say x and y, nothing can be determined, since for each possible secret K, there is a (unique) block in $\mathcal{B}_K$ that contains x and y.

1.4 Key Distribution Patterns

Suppose there are n users in a large network, say $P_1, P_2, \ldots, P_n$, where every pair of users might wish to communicate in a secure way. We assume that a conventional (i.e., symmetric or private key) cryptosystem is being used for secret communication. Then the problem can be solved by letting a trusted authority generate and distribute a secret key for each pair of users. The basic problem with this simpleminded solution is that it involves $\binom{n}{2}$ keys and requires each user to store $n - 1$ keys. In a large network, this could give rise to considerable key storage problems.

The number of keys that must be stored by each user can be considerably reduced under some weaker security conditions. We describe one approach based on combinatorial principles.

A *Key Distribution Pattern* (KDP) with parameters (n, v, m) is a pair $(X, \mathcal{B})$, where X is a set of v keys, say $\{K_1, K_2, \ldots, K_v\}$, and $\mathcal{B}$ is a set of n m-subsets of X, say $\{B_1, B_2, \ldots, B_n\}$.

KDPs are used as follows. Each one of the n users is given a set of m keys. To be precise, the user P_i is given the set of keys B_i. The key to be used by users P_i and P_j to communicate with one another is computed by applying a special function (e.g., a one-way function or a resilient function) to the set of common keys held by the two users (i.e., to $B_i \cap B_j$).

A KDP is said to be *w-collusion resistant* if the key used for communication between any two users, say P_i and P_j, cannot be computed by a coalition of w other malicious users who pool their sets of keys. More formally, the property can be stated as follows: For every pair of users P_i, P_j and for every set of w users C such that $C \cap \{P_i, P_j\} = \emptyset$, we have

$$\left(B_i \bigcap B_j\right) \setminus \left(\bigcup_{P_\ell \in C} B_\ell\right) \neq \emptyset.$$

We now discuss how w-collusion resistant key distribution patterns can be constructed from certain 3-designs.

1.15 Theorem

Suppose a 3-(n, k, λ_3) design exists such that $\lambda_2 > w\lambda_3$. Then, there exists a (n, b, λ_1) w-collusion resistant key distribution pattern.

The proof is simple: We use the *dual* of the 3-(n, k, λ_3) design as the key distribution pattern. The number of users in the KDP is n (the number of points in the 3-design). The number of keys in the KDP is b (the number of blocks in the 3-design). In the 3-design, each point occurs in exactly λ_1 blocks, and so each user gets λ_1 keys in the KDP.

By definition, we have for any $1 \le i < j \le n$

$$|B_i \cap B_j| = \lambda_2,$$

and, further,

$$|B_i \cap B_j \cap B_\ell| = \lambda_3$$

for every $P_\ell \in C$. Since $\lambda_2 > w\lambda_3$, it follows immediately that the key distribution pattern is w-collusion resistant.

1.5 Correlation-Immune and Resilient Functions

We begin with some definitions.

Let S be a set, $|S| = q$. Let $n \ge m \ge 1$ be integers and suppose $f : S^n \longrightarrow S^m$ is a function.

1. f is said to have *balanced input* if every possible input vector occurs with equal probability $1/q^n$.
2. f is said to be an (n, m, t, q) - *correlation immune* function if the probability of obtaining an output m-tuple $(y_1, \ldots, y_m)$ is exactly the same when the input is balanced as it is when the values of t inputs are fixed and the remaining $n - t$ inputs are chosen independently at random.
3. f is said to have *balanced output* if every possible output m-tuple occurs with equal probability $1/q^m$.
4. f is said to be an (n, m, t, q) - *resilient function* if it is t th-order correlation-immune and has balanced output.

Here are some examples of resilient functions, where arithmetic is to be performed in $GF(q)$:

1.16 Example

1. $m = 1$, $t = n - 1$. Define $f(x_1, \ldots, x_n) = x_1 + \ldots + x_n$.
2. $m = n - 1$, $t = 1$. Define $f(x_1, \ldots, x_n) = (x_1 + x_2, x_2 + x_3, \ldots, x_{n-1} + x_n)$.
3. $m = 2$, $n = 3h$, $t = 2h - 1$, and $q = 2$. Define

$$f(x_1, \ldots, x_{3h}) = (x_1 + \ldots + x_{2h}, x_{h+1} + \ldots + x_{3h}).$$

The following theorems describe the connection between correlation immune functions, resilient functions and orthogonal arrays.

1.17 Theorem An (n, m, t, q) - resilient function is equivalent to a large set of orthogonal arrays $LOA_{q^{n-m-t}}(t, n, q)$.

To state the next theorem, we need a definition. A *generalized large set of orthogonal arrays* $GLOA[q^m]_{\lambda_1, \ldots, \lambda_{q^m}}(t, n, q)$ consists of q^m simple arrays $OA_{\lambda_i}(t, n, q)$ whose union contains all possible n-tuples among its rows, so

$$\sum_{i=1}^{q^m} \lambda_i = q^{n-t}.$$

1.18 Theorem An (n, m, t, q) - CI function is equivalent to a generalized large set of orthogonal arrays $GLOA[q^m]_{\lambda_1, \ldots, \lambda_{q^m}}(t, n, q)$.

We outline the proof of the first theorem. If we start with (n, m, t, q) - resilient function, for any $y \in [GF(q)]^m$, the inverse image $f^{-1}(y)$ yields an $OA_{q^{n-m-t}}(t, n, q)$ and the q^m orthogonal arrays thus obtained form a large set of orthogonal arrays $LOA_{q^{n-m-t}}(t, n, q)$. The converse is also straightforward.

1.19 Example As an illustration, consider the following binary resilient function:

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = (x_1 + x_2 + x_3 + x_4, x_3 + x_4 + x_5 + x_6),$$

where addition is modulo 2. This is a $(6, 2, 3, 2)$-resilient function, and by the above theorem, it is equivalent to a large set of $OA_2(3, 6, 2)$. There are four orthogonal arrays in the large set, one of which is obtained from $f^{-1}(0, 0)$:

$$\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{array} \qquad \begin{array}{cccccc} 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{array}$$

The following theorem gives a construction for resilient functions from linear codes.

1.20 Theorem If there is a linear k-dimensional code of length n over $GF(q)$ having minimum Hamming distance d, then there is a $(n, m, d-1, q)$-resilient function.

1.6 See Also

§??	for information on perpendicular arrays
§??	for information on orthogonal arrays of strength 2
§??	for information on orthogonal arrays of strength $t > 2$
§??	for information on 2-designs
§??	for information on 3-designs
§??	for information on t-designs, $t > 3$
§??	for information on linear codes
[8]	Shannon's classic paper on secrecy systems
[15]	for a survey on combinatorial aspects of cryptography
[10]	for a survey on authentication codes
[11, 14]	for surveys on secret sharing schemes
[5]	the paper that introduced authentication codes
[12, 13]	for connections between authentication codes and designs
[7, 2]	the papers that introduced threshold schemes
[16]	for the connection between anonymous threshold schemes and designs
[6]	the paper that introduced key distribution patterns
[9]	the paper that introduced correlation immune functions
[3, 4]	the papers that introduced resilient functions
[1]	for recent results on resilient functions

References

[1] J. Bierbrauer, K. Gopalakrishnan and D. R. Stinson. Orthogonal Arrays, Resilient Functions, Error Correcting Codes and Linear Programming Bounds. Submitted to *SIAM J. Discrete Mathematics*.

[2] G. R. Blakley. Safeguarding cryptographic keys. *AFIPS Conference Proceedings*, **48** (1979), 313–317.

[3] C. H. Bennett, G. Brassard, and J. M. Robert. How to Reduce Your Enemy's Information. *Lecture Notes Comput. Sci.*, **218** (1986), 468–476.

[4] B. Chor, O. Goldreich, J. Håstad, J. Friedman, S. Rudich, and R. Smolensky. The Bit Extraction Problem or t-Resilient Functions. *IEEE Symposium on Foundations of Computer Science*, **26** (1985), 396–407.

[5] E. N. Gilbert, F. J. MacWilliams, and N. J. A. Sloane. Codes which detect deception. *Bell System Tech. Journal*, **53** (1974), 405–424.

[6] C. J. Mitchell and F. C. Piper. Key Storage in Secure Networks. *Discrete Applied Mathematics*, **21** (1988), 215–228.

[7] A. Shamir. How to share a secret. *Commun. of the ACM*, **22** (1979), 612–613.

[8] C. E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, **28** (1949), 656–715.

[9] T. Siegenthaler. Correlation immunity of nonlinear combining functions for cryptographic applications. *IEEE Trans. Inform. Theory*, **30** (1984), 776–780.

[10] G. J. Simmons. A survey of information authentication. In *Contemporary Cryptology, The Science of Information Integrity*, IEEE Press, 1992, pp. 379–420.

[11] G. J. Simmons. An Introduction to Shared Secret and/or Shared Control Schemes and Their Application. In *Contemporary Cryptology, The Science of Information Integrity*, IEEE Press, 1992, pp. 441–500.

[12] D. R. Stinson. The combinatorics of authentication and secrecy codes. *Journal of Cryptology*, **2** (1990), 23–49.

[13] D. R. Stinson. Combinatorial characterizations of authentication codes. *Designs, Codes and Cryptography*, **2** (1992), 175–187.

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

[15] D. R. Stinson. Combinatorial Designs and Cryptography. In *Surveys in Combinatorics , 1993*, London Math. Soc. Lecture Notes, **287** (1993), 257–287.

[16] D. R. Stinson and S. A. Vanstone. A combinatorial approach to threshold schemes. *SIAM J. on Discrete Math.*, **1** (1988), 230–237.