Queen Mary
University of London

MAS 335 **Cryptography**

Notes 11: Secret sharing **Spring 2008**

This section is primarily about secret sharing, a topic closely related to cryptography. There are also some remarks about other topics: other cryptographic protocols, and other types of attack on ciphers.

Secret sharing

The President of the Commercial Bank of Nod is the only person who holds a secret password which opens the bank vault. He realises that he can't always be around, and sometimes it is necessary to open the vault in his absence. But he doesn't trust any of his employees with the password. So he wants to give each of the two Vice-Presidents of the bank some partial information, so that only if the two of them combine their information can they open the vault. How can he do this?

He could simply give half the password to each Vice-President. But then there is a risk that one of the Vice-Presidents will guess the other half of the password: this is much easier than guessing the whole password. He wants a method where the information given to each Vice-President is no help to him in guessing the password on his own.

This is not difficult. He can simply encrypt the password, and give one Vice-President the ciphertext and the other the key. Together they can decrypt the password and open the vault. But if the cipher is a secure one such as a one-time pad, one Vice-President alone cannot break it, or even get any information about it (by Shannon's Theorem).

Slightly more formally, suppose that the password is a string over an alphabet A with q symbols. Let L be a $q \times q$ Latin square whose rows and columns are labelled by A, and whose entries are symbols from A. Suppose that $z = z_1 \ldots z_n$ is the password. Choose any random string $a = a_1 \ldots a_n$ of symbols of A, and let $b = b_1 \ldots b_n$ be the string for which $a_i \oplus b_i = z_i$ for $i = 1, \ldots, n$, where $s \oplus t$ is the symbol in row s and column t of the square. (This is slightly different from the way we did it before but the difference is immaterial.)

As usual, we write $z = a \oplus b$ to mean coordinatewise operation, that is, $z_i = a_i \oplus b_i$ for $i = 1, \dots, n$.

For example, let $A = \{0, \dots, q-1\}$ be the set of integers mod q, and L is the addition table mod q (so that $s \oplus t = s + t \bmod q$).

This example can easily be extended to the case where there are k Vice-Presidents, and it is required that only all k acting together can open the vault. Let us suppose that the Latin square is the addition table mod q. In this case, the jth Vice-President is given the information $a^{(j)} = a_1^{(j)} \dots a_n^{(j)}$, where

$$z = a^{(1)} \oplus a^{(2)} \oplus \cdots \oplus a^{(k)}.$$

(For an arbitrary Latin square the method is the same, but we have to be careful about the order we do the additions.)

Not only is it true that any $k-1$ of the Vice-Presidents cannot work out the password; they cannot get any information at all about it. For example, suppose that the first $k-1$ Vice-Presidents co-operate. They can calculate

$$b = a^{(1)} \oplus a^{(2)} \oplus \cdots \oplus a^{(k-1)}.$$

Now

$$b \oplus a^{(k)} = z,$$

but because of the Latin square property, in each row every symbol occurs once, so without knowledge of $a^{(k)}$ all strings are equally likely!

We can extend this idea still further with a definition as follows. Let k and t be positive integers with $k > t$, and let A be an alphabet of q symbols. A (k,t) *orthogonal array* over A is defined to be an array M with k rows and q^t columns with entries from A, having the following property:

> Given any t rows of M, and any t elements $a_1, \dots, a_t$ of A, there is exactly one column of A in which the entries $a_1, \dots, a_t$ occur (in that order) in the t chosen rows.

The numbers k and t are called the *degree* and *strength* respectively of the orthogonal array. The number of columns must be q^t, since this is the number of choices of a t-tuple $(a_1, \dots, a_t)$.

Recall that, for a Latin square with symbol set $\{1, \dots, n\}$, we constructed an array with three rows and q^2 columns, where the three entries of each column give the row number, column number, and symbol contained in a cell of the square. The defining properties of a Latin square translate into the fact that this is an orthogonal array of degree 3 and strength 2. (A row and column uniquely determine a symbol; a row and symbol uniquely determine a column; and a column and symbol uniquely determine a row.)

A (k,t) *secret sharing scheme* is a scheme in which each of k individuals is given a member of a set S in such a way that any t of the individuals acting together can determine the identity of a secret member s of S, but no $t-1$ individuals can get any information about s.

Theorem 25 *From an orthogonal array of degree k and strength t over A, we can construct a $(k-1,t)$ secret sharing scheme over the set A^n of strings of length n of elements of A.*

The construction works as follows. We can take $n=1$, since to "encode" a string we simply deal with its characters one at a time.

Suppose that M is an orthogonal array of degree k and strength t over A. The array M is regarded as public.

Now M has q^t columns. Exactly q^{t-1} of these have the property that the secret s appears in the last row. Choose one of these columns at random, and give the entry in its ith row to the ith individual in the secret-sharing scheme for $i=1,\ldots,k-1$.

Now by the properties of an orthogonal array, any t of the individuals can, by pooling their information, determine the chosen column of M, and hence its last entry, which is the secret. However, the information held by any $t-1$ individuals only determines a set of q columns, with the property that each symbol occurs in the last row of precisely one of these columns. So the individuals concerned can obtain no information about the secret.

Thus, a Latin square gives a $(2,2)$ secret sharing scheme, and we have seen that we can use it to construct a (k,k) secret sharing scheme for any k.

There is a construction for making smaller orthogonal arrays from larger ones. Suppose that S is an orthogonal array of degree k and strength t.

- Deleting any row of the array gives an orthogonal array of degree $k-1$ and strength t, assuming that $t<k$. This is clear: any t rows of the remaining array contain t given symbols in exactly one column.

- Let a_0 be a fixed symbol of the alpabet. Select those columns which have a_0 in the last row (there are q^{t-1} of these, since each symbol occurs equally often in the last row), and delete the remaining columns. Then delete the last row. The resulting $(k-1)\times q^{t-1}$ array is an orthogonal array of degree $k-1$ and strength $t-1$. For given $t-1$ rows of this array, there is a unique column of the original array which has $t-1$ prescribed entries in these rows and a_0 in the last row.

These constructions only produce smaller arrays from larger ones; we need a construction to give us some orthogonal arrays to start with. This a bit harder. Here is a very nice construction due to K. A. Bush:

Theorem 26 *Let q be a prime power, and t a positive integer less than $q+1$. Then there exists an orthogonal array of degree $q+1$ and strength t over an alphabet of size q (and hence a (q,t) secret sharing scheme over an alphabet of size q).*

In this case, the alphabet is the finite field $\mathrm{GF}(q)$ with q elements. The array is constructed as follows.

Consider polynomials of degree $t-1$. Each such polynomial has the form

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

where $a_0, a_1, \ldots, a_{t-1} \in \mathrm{GF}(q)$. So there are q choices for each of the t coefficients, and hence q^t polynomials.

From any polynomial $f(x)$, we construct a column of length $q+1$ as follows. If the elements of $\mathrm{GF}(q)$ are numbered $u_1, \ldots, u_q$, we put $f(u_i)$ in the ith row, for $i = 1, \ldots, q$. In the $(q+1)$st row, we put the leading coefficient a_{t-1} of $f(x)$.

This gives an array with $q+1$ rows and q^t columns. It remains to show that it is an orthogonal array of strength t. Suppose we seek a column in which rows $i_1, \ldots, i_t$ contain entries $z_1, \ldots, z_t$ respectively.

Suppose first that none of these rows is the $(q+1)$st. To ease notation, we put $u_{i_j} = v_j$ for $j = 1, \ldots, t$. Then we have to show that there is a unique polynomial $f(x)$ of degree at most $t-1$ such that it takes prescribed values at t given points, namely

$$f(v_j) = z_j, \qquad j = 1, \ldots, t.$$

This is true in general; the method for finding the polynomial is known as *Lagrange interpolation*. In the case of a finite field, it can be proved by simple counting. We give this argument, and then the general proof (which has the advantage of being constructive).

First, we observe that there is at most one polynomial of degree $\leq t-1$ taking these values. For if f and g were two such polynomials, then $f-g$ would be zero at each point $v_1, \ldots, v_t$, contradicting the fact that a polynomial cannot have more roots than its degree. (This part of the argument works for any field.)

Now, there are q^t choices of the t values $z_1, \ldots, z_t$, and there are q^t choices of the coefficients of the polynomial

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

so each list of values must be realised by a polynomial.

The constructive method works as follows. Let

$$g_i(x) = \prod_{j \neq i} \frac{(x - v_j)}{(v_i - v_j)}.$$

Then we have $g_i(v_i) = 1$, and $g_i(v_j) = 0$ for $j \neq i$. Hence

$$f(x) = \sum_{i=1}^{t} z_i g_i(x)$$

satisfies $f(v_i) = z_i$ for $i = 1, \ldots, t$.

Now suppose that one of the rows (say the last) is the $(q+1)$st. Then in place of what went before, the last equation is now $a_{t-1} = z_t$. This equation determines a_{t-1}, and so we have to interpolate a polynomial h of degree $\leq t-2$ taking the other $t-1$ values

$$h(v_i) = z_i - a_{t-1} v_i^{t-1}, \qquad i = 1, \ldots, t-1.$$

By the same argument as before, there is a unique such polynomial.

The implementation of this secret-sharing scheme is remarkably simple. The President takes the secret password to be the coefficient of x^{t-1} in the polynomial, and chooses the coefficients of lower-degree terms at random. Then he evaluates the polynomial on the elements of the field, and gives one value to each Vice-President.

Any t of the Vice-Presidents can now use Lagrange interpolation, as described above, to find the unique polynomial of degree at most $t-1$ taking the values they have been given. Its leading coefficient is the secret. On the other hand, fewer than t Vice-Presidents can gain no information at all about the secret.

Example Figure 1 is the orthogonal array of degree 4 and strength 3 constructed by the above method. The array has been transposed for convenience in printing. We take all polynomials of degree at most 2 over GF(3) $= \{0,1,2\}$. The components of the 4-tuple are $f(0)$, $f(1)$, $f(2)$, and the coefficient of x^2 in $f(x)$

Run your fingers down any three columns of the array on the right, and you should find that each of the $3^3 = 27$ possible triples occur exactly once.

Remark Bush's orthogonal arrays are known, in different terminology, as *Reed-Solomon codes*, and are used for error correction in CD players.

Polynomial	4-tuple
0	0000
1	1110
2	2220
x	0120
$x + 1$	1200
$x + 2$	2010
$2x$	0210
$2x + 1$	1020
$2x + 2$	2100
x^2	0111
$x^2 + 1$	1221
$x^2 + 2$	2001
$x^2 + x$	0201
$x^2 + x + 1$	1011
$x^2 + x + 2$	2121
$x^2 + 2x$	0021
$x^2 + 2x + 1$	1101
$x^2 + 2x + 2$	2211
$2x^2$	0222
$2x^2 + 1$	1002
$2x^2 + 2$	2112
$2x^2 + x$	0012
$2x^2 + x + 1$	1122
$2x^2 + x + 2$	2202
$2x^2 + 2x$	0102
$2x^2 + 2x + 1$	1212
$2x^2 + 2x + 2$	2022

Figure 1: An orthogonal array

Other protocols

Session keys

Public-key cryptography is slower than cryptography based on a shared secret key. So many systems, including PGP, have an initial round where a public-key cipher is used to share a secret key between the two participants of a session. The key is used only for that communication session.

The simplest way to do this is a modification of the Diffie–Hellman method. It has the advantage that the key itself is not transmitted, even in enciphered form.

Alice and Bob share a prime number p and a primitive root g mod p. (They must assume that Eve knows p and g as well.) Now Alice choses a number a in the range $\{0,\ldots,p-2\}$ and Bob chooses b in the same range. Alice computes g^a mod p and sends it to Bob; Bob computes g^b mod p and sends it to Alice. Now each of them can compute $(g^a)^b = (g^b)^a$ mod p; this is the session key.

To obtain the key, Eve knows g^a and g^b, but needs either a or b to proceed further; so she needs to solve a discrete logarithm problem. Since a new key can be chosen for each session, Eve cannot pre-compute the discrete logarithm of a public key as in the case of El-Gamal.

Note that, as opposed to the protocol described before, this method requires only two, rather than three, transmissions, and these are asynchronous (that is, they can occur in either order).

And more ...

Protocols for many other tasks have been derived. For example, Alice can send Bob a message which he has a 50% chance of being able to decrypt, and Alice herself doesn't know whether or not Bob can decrypt it. Similarly, she can send him a message which allows him to learn one or other of two secrets, so that Alice does not know which secret Bob has learned. Bob may construct a smart card which knows his secret key, and can prove that it knows it, but without revealing the secret key.

Fanciful as these may sound, they have been suggested to solve real practical problems. The last protocol, for example, has been proposed by Shamir as the basis for an electronic passport.

Other kinds of attack

For the most part, we have imagined Eve as just a snooper who intercepts a message from Alice to Bob and must be prevented from knowing its contents. There are more active roles that she can play. Here are a few examples. There has been a lot of work on deciding whether the ciphers we have discussed are secure against this kind of attack. Sometimes we must imagine that Eve is someone who works in Alice's or Bob's organisation, or someone who has complete control of the communication channel between them.

- Eve may have access to some ciphertexts from Alice to Bob together with the corresponding plaintext. Does this help her break future messages?
- Eve may, in some circumstances, be able to persuade Alice to encipher messages of Eve's choosing. Carefully-chosen messages may give more information than arbitrary messages.
- Eve may be able to impersonate Alice to a greater or lesser degree. For example, she can certainly send Bob a message claimimg to come from Alice, encrypted with Bob's public key. Alice can foil this by signing or authenticating her messages; we have seen how to do this in both RSA and El-Gamal. Even in this case, Eve may be able to send Bob some previously-intercepted ciphertexts instead ot the current ciphertext that Alice wants to send.
- Alice may later wish to repudiate a message she has sent to Bob, claiming that it was a forgery from Eve. If it is signed (and the signature includes a date and time), this should not be possible; but it seems difficult to prevent Alice from claiming that her private key has been obtained illicitly by Eve.