Elliptic Curves

Jang Schiltz

Centre Universitaire de Luxembourg
Séminaire de Mathématiques
162A, avenue de la Faïencerie
L-1511 Luxembourg
Luxembourg
E-mail:schiltzj@cu.lu

1 Finite fields

1.1 Generalities

Definition 1.1 *A finite field is an algebraic system consisting of a finite set F together with two binary operations $+$ and $\times$, defined on F, satisfying the following axioms:*

- *F is an abelian group with respect to "+"*
- *$F \backslash \{0\}$ is an abelian group with respect to "×"*
- *distributive: for all x, y and z in F we have:*

$$x \times (y + z) = (x \times y) + (x \times z)$$
$$(x + y) \times z = (x \times z) + (y \times z).$$

Definition 1.2 *The order of a finite field F is the number of elements in F. Is is denoted by $|F|$.*

Theorem 1.3 *There exists a finite field of order q if and only if q is a prime power. In addition, if q is a prime power, then there is essentially only one finite field of order q; this field is denoted by $\mathbb{F}_q$.*

Definition 1.4 *If $q = p^m$, where p is a prime and m a positive integer, then p is called the characteristic of $\mathbb{F}_q$ and m is called the extension degree of $\mathbb{F}_q$.*

1.2 The finite field $\mathbb{F}_{p^m}$

Let p be a prime number. The finite field $\mathbb{F}_q$, with $q = p^m$, called a characteristic p finite field, can be viewed as a vector space of dimension m over $\mathbb{F}_p$. That is, there exists a set of m elements $\{\alpha_0, \alpha_1, ..., \alpha_{m-1}\}$ in $\mathbb{F}_{p^m}$ such that each $a \in \mathbb{F}_{p^m}$ can be written uniquely in the form

$$a = \sum_{i=0}^{m-1} a_i \alpha_i, \text{ where } \alpha_i \in 0, 1, ..., p-1.$$

The set $\{\alpha_0, \alpha_1, ..., \alpha_{m-1}\}$ is called a basis of $\mathbb{F}_{p^m}$ over $\mathbb{F}_p$. We can then represent a as the bit string $[a_{m-1}a_{m-2}...a_0]$. Addition of field elements is performed by bitwise XOR-ing the vector representations. The multiplication rule depends on the basis selected.

There are many different bases of $\mathbb{F}_{p^m}$ over $\mathbb{F}_p$. Some bases lead to more efficient software or hardware implementations of the arithmetic in $\mathbb{F}_{p^m}$ than other bases. The two most commonly used kinds of bases are polynomial bases and normal bases.

1.2.1 Polynomial basis representation

Definition 1.5 *Let $f(x) = x^m + \sum_{i=0}^{m-1} f_i x^i$ (where $f_i \in 0, 1, ..., p-1$) be an irreducible polynomial of degree m over $\mathbb{F}_p$, that is, $f(x)$ cannot be factored as a product of two polynomials over $\mathbb{F}_p$, each of degree lesser than m; $f(x)$ is called the reduction polynomial. For each reduction polynomial there exists a polynomial basis representation.*

Field elements. The finite field $\mathbb{F}_{p^m}$ is comprised of all polynomials over $\mathbb{F}_p$ of degree less than m:

$$\mathbb{F}_{p^m} = \left\{a_{m-1}x^{m-1} + ... + a_1 x + a_0 : a_i \in \{0, 1, ..., p-1\}\right\}.$$

The field element $a_{m-1}x^{m-1} + ... + a_1 x + a_0$ is usually denoted by the bit string $[a_{m-1}...a_1\, a_0]$ of length m, so that

$$\mathbb{F}_{p^m} = \{[a_{m-1}...a_1\, a_0] : a_i \in \{0, 1, ..., p-1\}\}.$$

Definition 1.6 *The following operations are defined on the elements of $\mathbb{F}_{p^m}$ when using a polynomial representation with reduction polynomial $f(x)$. Assume that $a = [a_{m-1}...a_1 a_0]$ and $b = [b_{m-1}...b_1 b_0]$.*

- *Addition: $a + b = c = [c_{m-1}...c_1 c_0]$, where $c_i \equiv a_i + b_i (mod\ p)$.*
- *Multiplication: $a \cdot b = c = [c_{m-1}...c_1 c_0]$, where $c(x) = \sum_{i=0}^{m-1} c_i x^i$ is the remainder of the division of the polynomial $(\sum_{i=0}^{m-1} a_i x^i)(\sum_{i=0}^{m-1} b_i x^i)$ by $f(x)$ over $\mathbb{F}_p$.*
- *Inversion: if a is a non-zero element in $\mathbb{F}_{p^m}$, the inverse of a, denoted by a^{-1}, is the unique element $c \in \mathbb{F}_{p^m}$ for which $a \cdot c = 1$.*

Example 1.7 *Let $p = 2$, i.e. $\mathbb{F} = \mathbb{F}_{2^m}$ is a binary finite field and $f(x) = x^3 + x + 1$. This polynomial is irreducible. Indeed, $f(0) = f(1) = 0$, hence f has no zero's on $\mathbb{F}_2$. Since $f(x)$ is a polynomial of degree 3, the elements of $\mathbb{F}_{2^3}$ can be written as $a_2x^2 + a_1x + a_0$, with $a_i \in \mathbb{F}_2$ for $i = 1, 2, 3$.*

Moreover,

$$\begin{aligned}
&(a_2x^2 + a_1x + a_0)(b_2x^2 + b_1x + b_0) \\
&\qquad = a_2b_2x^4 + (a_2b_1 + a_1b_2)x^3 + (a_2b_0 + a_1b_1 + a_0b_2)x^2 + (a_1b_0 + a_0b_1)x + a_0b_0 \\
&\qquad = (a_2b_2 + a_2b_0 + a_1b_1 + a_0b_2)x^2 + (a_2b_2 + a_2b_1 + a_1b_2 + a_1b_0 + a_0b_1)x \\
&\qquad\quad + a_2b_1 + a_1b_2 + a_0b_0,
\end{aligned}$$

since

$$x^3 \equiv x + 1 \pmod{x^3 + x + 1}$$

and consequently

$$x^4 \equiv x^2 + x \pmod{x^3 + x + 1}$$

Hence,

$$\begin{aligned}
[a_2a_1a_0] \cdot [b_2b_1b_0] = [&a_2b_2 + a_2b_0 + a_1b_1 + a_0b_2 \\
&a_2b_2 + a_2b_1 + a_1b_2 + a_1b_0 + a_0b_1 \;\; a_2b_1 + a_1b_2 + a_0b_0].
\end{aligned}$$

Example 1.8 *(*A polynomial basis representation of the finite field $\mathbb{F}_{2^4}$*) Let $f(x) = x^4 + x + 1$ be the reduction polynomial. Then the 16 elements of $\mathbb{F}_{2^4}$ are:*

0	[0000]	x^3	[1000]
1	[0001]	$x^3 + 1$	[1001]
x	[0010]	$x^3 + x$	[1010]
$x + 1$	[0011]	$x^3 + x + 1$	[1011]
x^2	[0100]	$x^3 + x^2$	[1100]
$x^2 + 1$	[0101]	$x^3 + x^2 + 1$	[1101]
$x^2 + x$	[0110]	$x^3 + x^2 + x$	[1110]
$x^2 + x + 1$	[0111]	$x^3 + x^2 + x + 1$	[1111]

Examples of the arithmetic operations in $\mathbb{F}_{2^4}$ are:
$[1101] + [1001] = [0100]$.
$[1101] \cdot [1001] = [1111]$, *since* $(x^3 + x^2 + 1) \cdot (x^3 + 1) = x^6 + x^5 + x^2 + 1 \equiv x^3 + x^2 + x + 1 \bmod (x^4 + x + 1)$.
$[1101]^{-1} = [0100]$.

The element $\alpha = [0100]$ is a generator of $\mathbb{F}_{2^4}^*$ since its order is 15 as the following *calculations show:*

$$\begin{array}{ccc}
\alpha^1 = [0010] & \alpha^2 = [0100] & \alpha^3 = [1000] \\
\alpha^4 = [0011] & \alpha^5 = [0110] & \alpha^6 = [1100] \\
\alpha^7 = [1011] & \alpha^8 = [0101] & \alpha^9 = [1010] \\
\alpha^{10} = [0111] & \alpha^{11} = [1110] & \alpha^{12} = [1111] \\
\alpha^{13} = [1101] & \alpha^{14} = [1001] & \alpha^{15} = [0001]
\end{array}$$

Selecting a reduction polynomial. The following procedure is commonly used to choose a reduction polynomial: if an irreducible trinomial $x^m + x^k + 1$, where $1 \leq k \leq m-1$ exists over $\mathbb{F}_2$, then the reduction polynomial $f(x)$ is chosen to be the irreducible trinomial with the lowest-degree middle term x^k. If no irreducible trinomial exists, then select instead a pentanomial $x^m + x^{k_3} + x^{k_2} + x^{k_1} + 1$, where $1 \leq k_1 < k_2 < k_3 \leq m-1$, such that k_3 has the minimal value, the value of k_2 is minimal for the given k_3 and k_1 is minimal for the given k_3 and k_2.

1.2.2 The finite field $\mathbb{F}_p$

Definition 1.9 *Let p be a prime number. The finite field $\mathbb{F}_p$, called a prime field consists of the set of integers*

$$\{0, 1, ..., p-1\}$$

with the following arithmetic operations:

- *addition: if $a, b \in \mathbb{F}_p$, then $a \cdot b = r$, where r is the remainder of the division of $a+b$ by p and $0 \leq r \leq p-1$. This operation is called addition modulo p.*
- *multiplication: if $a, b \in \mathbb{F}_p$, then $a \cdot b = s$, where s is the remainder of the division of $a \cdot b$ by p and $0 \leq r \leq p-1$. This operation is called multiplication modulo p.*
- *inversion: if a is a non-zero element in $\mathbb{F}_p$, the inverse of a modulo p, denoted a^{-1}, is the unique integer $c \in \mathbb{F}_p$ for which $a \cdot c = 1$.*

Example 1.10 *(The finite field $\mathbb{F}_{23}$) The elements of $\mathbb{F}_{23}$ are $1, 2, ..., 22$. Examples of the arithmetic operations in $\mathbb{F}_{23}$ are: (1)* $12 + 20 = 9$*; (2)* $8 \cdot 9 = 3$*; (3)* $8^{-1} = 3$.

There are certain primes p for which the modular reduction can be computed very efficiently. For example, let p be the prime $2^{192} - 2^{64} - 1$. To reduce a positive integer $n < p^2$, write

$$n = \sum_{j=0}^{5} A_j \cdot 2^{64j}.$$

Then,

$$n \equiv T + S_1 + S_2 + S_3 \pmod{p},$$

where

$$\begin{array}{ll} T &= A_2 \cdot 2^{128} + A_1 \cdot 2^{64} + A_0 \\ S_1 &= \phantom{A_4 \cdot 2^{128} +} A_3 \cdot 2^{64} + A_3 \\ S_2 &= A_4 \cdot 2^{128} + A_4 \cdot 2^{64} \\ S_3 &= A_5 \cdot 2^{128} + A_5 \cdot 2^{64} + A_5. \end{array}$$

2 Elliptic curves

2.1 Elliptic curves over $\mathbb{F}_p$

Definition 2.1 *Let $p > 3$ be an odd prime and let $a, b \in \mathbb{F}_p$ satisfy $4a^3 + 27b^2 \not\equiv 0$ (mod p). Then an elliptic curve $E(\mathbb{F}_p)$ over $\mathbb{F}_p$ defined by the parameters $a, b \in \mathbb{F}_p$ consists of the set of solutions or points $P = (x, y)$ for $x, y \in \mathbb{F}_p$ to the equation:*

$$y^2 = x^3 + ax + b$$

together with a special point $\mathcal{O}$ called the point at infinity. For a given point $P = (x_P, y_P)$, x_p is called the x-coordinate of P and y_P is called the y-coordinate of P. $\mathbb{F}_p$ is called a prime finite field.

On an elliptic curve $E(\mathbb{F}_p)$ over a field $\mathbb{F}_p$ can be defined a binary operation $+$ as follows:

1. $P + \mathcal{O} = \mathcal{O} + P = P$, for all $P \in E(\mathbb{F}_p)$.

2. If $P = (x, y) \in E(\mathbb{F}_p)$, then $(x, y) + (x, -y) = \mathcal{O}$. The point $(x, -y) \in E(\mathbb{F}_p)$ is denoted $-P$ and is called the negative of P.

3. Let $P = (x_1, y_1) \in E(\mathbb{F}_p)$ and $Q = (x_2, y_2) \in E(\mathbb{F}_p)$, where $P \neq \pm Q$. Then $P + Q = (x_3, y_3)$, where

$$x_3 = \lambda^2 - x_1 - x_2,\ y_3 = \lambda(x_1 - x_3) - y_1, \text{ and } \lambda = \frac{y_2 - y_1}{x_2 - x_1}.$$

4. Let $P = (x_1, y_1) \in E(\mathbb{F}_p)$. Then $P + P = 2P = (x_3, y_3)$, where

$$x_3 = \lambda^2 - 2x_1,\ y_3 = \lambda(x_1 - x_3) - y_1, \text{ and } \lambda = \frac{3x_1^2 + a}{2y_1}.$$

   This operation is called the doubling of a point.

Thus $\big(E(\mathbb{F}_p), +\big)$ forms an abelian group.

2.2 Elliptic curves over $\mathbb{F}_{2^m}$

Definition 2.2 *A (non-supersingular) elliptic curve $E(\mathbb{F}_{2^m})$ over $\mathbb{F}_{2^m}$ defined by the parameters $a, b \in \mathbb{F}_{2^m}, b \neq 0$ consists of the set of solutions or points $P = (x, y)$ for $x, y \in \mathbb{F}_{2^m}$ to the equation:*

$$y^2 + xy = x^3 + ax^2 + b$$

together with a special point $\mathcal{O}$ called the point at infinity. For a given point $P = (x_P, y_P)$, x_p is called the x-coordinate of P and y_P is called the y-coordinate of P. $\mathbb{F}_{2^m}$ is called a characteristic 2 finite field.

On an elliptic curve $E(\mathbb{F}_p)$ over a field $\mathbb{F}_p$ can be defined a binary operation $+$ as follows:

1. $P + \mathcal{O} = \mathcal{O} + P = P$, for all $P \in E(\mathbb{F}_{2^m})$.

2. If $P = (x, y) \in E(\mathbb{F}_{2^m})$, then $(x, y) + (x, -y) = \mathcal{O}$. The point $(x, -y) \in E(\mathbb{F}_{2^m})$ is denoted $-P$ and is called the negative of P.

3. Let $P = (x_1, y_1) \in E(\mathbb{F}_{2^m})$ and $Q = (x_2, y_2) \in E(\mathbb{F}_{2^m})$, where $P \neq \pm Q$. Then $P + Q = (x_3, y_3)$, where

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a,\ y_3 = \lambda(x_1 + x_3) + x_3 + y_1, \text{ and } \lambda = \frac{y_2 + y_1}{x_2 + x_1}.$$

4. Let $P = (x_1, y_1) \in E(\mathbb{F}_{2^m})$. Then $P + P = 2P = (x_3, y_3)$, where

$$x_3 = \lambda^2 + \lambda + a,\ y_3 = \lambda(x_1 + x_3) + x_3 + y_1, \text{ and } \lambda = x_1 + \frac{x_1}{y_1}.$$

   This operation is called the doubling of a point.

Thus $\big(E(\mathbb{F}_p), +\big)$ forms an abelian group.

2.2.1 Properties

The central operation of cryptographic schemes based on elliptic curve cryptography (ECC) is the elliptic scalar multiplication (operation analogue of the exponentiation in multiplicative groups)

Definition 2.3 *Given an integer k and a point P in a finite field $\mathbb{F}$, the elliptic scalar multiplication kP is the result of adding P to itself k times.*

Definition 2.4 *The order of a point P on an elliptic curve is the smallest positive integer r such that $rP = \mathcal{O}$. If k and l are integers, then $kP = lP$ if and only if $k \equiv l (mod\ r)$.*

Definition 2.5 *The number of points of $E(\mathbb{F})$, denoted by $\#E(\mathbb{F})$ is called the curve order of the curve $E(\mathbb{F})$.*

Definition 2.6 *The trace $Tr(\cdot)$ is the linear map from $\mathbb{F}_{2^m}$ to $\mathbb{F}_2$ defined by*

$$Tr(a) = \sum_{i=0}^{m-1} a^{2^i}.$$

Proposition 2.7 *Let E be an elliptic curve ofer a finite field $\mathbb{F}_q$. Then:*

- *Hasse's theorem states that $\#E(\mathbb{F}_q) = q + 1 - t$, where $|t| \leq 2\sqrt{q}$. That is, the number of points in $E(\mathbb{F}_q)$ is approximately q.*
- *If q is a power of 2, then $\#E(\mathbb{F}_q)$ is even. More specifically, $\#E(\mathbb{F}_q) \equiv 0 (mod\ 4)$ if $Tr(a) = 0$, and $\#E(\mathbb{F}_q) \equiv 2 (mod\ 4)$ if $Tr(a) = 1$.*
- *$E(\mathbb{F}_q)$ is an abelian group of rank 1 or 2. That is, $E(\mathbb{F}_q)$ is isomorphic to $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$, where n_2 divides n_1 and $q - 1$.*
- *If q is a power of two and $P = (x, y) \in E(\mathbb{F}_q)$ is a point of odd order, then the trace of the x-coordinate of all multiples of P is equal to the trace of the parameter a. That is, $Tr\big(x(kP)\big) = Tr(a)$ for each integer k.*

2.2.2 Koblitz curves

These curves, also known as binary anomalous curves, are elliptic curves over $\mathbb{F}_{2^m}$ with coefficients a and b either 0 or 1. Since it is required that $b \neq 0$, they are defined by the equations

$$E_0 : y^2 + xy = x^3 + 1 \text{ and } E_1 : y^2 + xy = x^3 + x^2 + 1.$$

Proposition 2.8 *If (x, y) is a point on $E_a, a = 0$ or 1, so is the point (x^2, y^2). Moreover, every point $P = (x, y) \in E_a$ satisfies the relation*

$$(x^4, y^4) + 2P = \mu \cdot (x^2, y^2),$$

where $\mu = (-1)^{1-a}$.

By using the Frobenius map over $\mathbb{F}_2 : \tau(x, y) = (x^2, y^2)$, this can be written as

$$\tau(\tau P) + 2P = \mu\tau P, \text{ for all } P \in E_a.$$

2.3 Elliptic curve cryptography

Unlike the ordinary discrete logarithm problem and the integer factorization problem, no subexponential-time algorithm is known for the elliptic curve discrete logarithm problem. For this reason, the strength-per-key-bit is substantially greater in an algorithm that uses elliptic curves. Thus, smaller parameters, but with equivalent levels of security, can be used with elliptic curve cryptosystems than with discrete logarithm systems.

2.4 Digital signature schemes

2.4.1 Generalities

Digital signature schemes are designed to provide the digital counterpart to handwritten signatures (and more). A digital signature is a number dependent on some secret known only to the signer (the signer's private key), and, additionnaly, on the contents of the message being signed. Signatures must be verifiable - if a dispute arises as to whether an entity signed a document, an unbiased third party should be able to resolve the matter equitably, without requiring access to the signer's private key. disputes may arise when a signer tries to repudiate a signature it did create, or when a forger makes a fraudulent claim.

We speak here about asymmetric digital signature schemes with an appendix. "Asymmetric" means that each entity selects a key pair consisting of a private key and a related public key. The entity maintains the secrecy of the private key that it uses for signing messages, and makes authentic copies of its public key available to other entities which use it to verify signatures. "Appendix" means that a cryptographic hash function is used to create a message digest of the message, and the signing transformation is applied to the message digest rather than to the message itelf.

Security. Ideally, a digital signature scheme should be existentially unforgeable under chosen-message attack. This notion of security asserts that an adversery who is able to obtain entity A's signatures for any message of its choice is unable to successfully forge A's signature on a single other message.

Applications. Digital signature schemes can be used to provide the following basic cryptographic services: data integrity (the assurance that data has not been altered by uauthorized or unknown means), data origin authentication (the assurance that the source of data is as claimed), and non-repudiation (the assurance that an entity cannot deny previous actions or commitments). Digital signature schemes are commonly used as primitives in cryptographic protocols that provide other services including entity authentication, authenticated key-transport and authenticated key agreement.

Classification. The digital signature schemes in use today can be classified according to the hard underlying mathematical problem which provides the basis for their security:

1. Integer factorization (IF) schemes, which base their security on the intracibility of the integer factorization problem. Examples of these include the RSA and Rabin signature schemes.

2. Discrete logarithm (DL) schemes, which base their security on the intractability of the (ordinary)discrete logarithm problem in a finite field. Examples of these include the ELGamal, Schnorr, DSA, and Nyberg-Ryppel signature schemes.

3. Elliptic curve (EC)schemes, which base their security on the intractability of the elliptic curve discrete logarithm problem.

2.4.2 The Digital Signature Algortihm (DSA)

The DSA was proposed in August 1991 by the U.S. National Institute of Standards and Technology (NIST) and was specified in a U.S. Governement Federal Information Processing Standard (FIPS 186) called the Digital Signature Standard (DSS). The DSA can be viewed as a variant of the ElGamal signature scheme. Its security is based on the intractability of the discrete logarithm problem in prime-order subgriuops of $\mathbb{Z}_p^*$.

DSA domain parameter generation. Domain parameters are generated for each entity in a particular security domain.

1. Select a 160-bit prime q and a 1024-bit prime p with the property that $q|p-1$.

2. (Select a generator g of the unique cyclic group of order q in $\mathbb{Z}_p^*$)

   Select an element $h \in \mathbb{Z}_p^*$ and compute $g = h^{(p-1)/q} \mod p$. (Repeat until $g \neq 1$.)

3. Domain parameters are p, q and g.

DSA key pair generation. Each entity A in the domain with domain parameters (p, q, g) does the following:

1. Select a random or pseudorandom integer x such that $1 \leq x \leq q-1$.

2. Compute $y = g^x \mod p$.

3. A's public key is y; A's private key is x.

DSA signature generation. To sign a message m, A does the following:

1. Select a random or pseudorandom integer $k, 1 \leq k \leq q-1$.

2. Compute $X = g^k \mod p$ and $r = X \mod q$. If $r = 0$ then go to step 1.

3. Compute $k^{-1} \mod q$.

4. compute e =SHA-1(m).

5. Compute $s = k^{-1}\{e + xr\} \mod q$. If $s = 0$ then go to step 1.

6. A's signature for the message m is (r, s).

DSA signature verification. To verify A's signature (r, s) on m, B obtains authentic copies of A's domain parameters (p, q, g) and a public key y and does the following:

1. Verify that r and s are integers in the interval $[1, q - 1]$.

2. Compute e =SHA-1(m).

3. Compute $w = s^{-1} \mod q$.

4. Compute $u_1 = ew \mod q$ and $u_2 = rw \mod q$.

5. Compute $X = g^{u_1}y^{u_2} \mod q$ and $v = X \mod q$.

6. Accept the signature if and only if $v = r$.

Security analysis. Since r and s are each integers less than q, DSA signatures are 320 bits in size. The security of the DSA relies on two distinct but related discrete logarithm problems. One is the discrete logarithm problem in $\mathbb{Z}_p^*$ where the number field sieve algorithm applies; this algorithm has a subexponential running time. More precisely, the expected running time of the algorithm is

$$O\left(\exp\Big(\big(c + o(1)\big)(\ln p)^{1/3}(\ln \ln p)^{2/3}\Big)\right), \tag{1}$$

where $c \approx 1,923$ and $\ln n$ denotes the natural logarithm function. If p is a 1024-bit prime, then the expression (1) represents an infeasible amount of computation; thus the DSA using a 1024-bit prime p is currently not vulnerable to this attack. The second discrete logarithm problem works to the base g in the subgroup of order q in $\mathbb{Z}_p^*$: given p, q, g and y, find x such that $y \equiv g^x(\mod p)$. For large p (e.g., 1024 bits), the best algorithm known for this problem is Pollard's rho method and takes about

$$\sqrt{\pi q/2} \tag{2}$$

steps. If $q \equiv 2^{160}$, then the expression (2) represents an infeasible amount of computation; thus the DSA is not vulnerable to this attack. However, note that there are two primary security parameters for DSA: the size of p and the size of q. Increasing one without a corresponding increase in the other will not result in an effective increase in security. Furthermore, an advance in algorithms for either one of the two discrete logarithm problems could weaken RSA.

Secure generation of parameters. In response to some criticisms received on the first draft, FIPS 186 specified a method for generating primes p and q "verifiably at random". This feature prevents an entity (e.g., a central authority generating domain parameters to be shared by a network of entities) from intentionally constructing "weak" primes p and q for which the discrete logarithm problem is relatively easy. FIPS 186 also specifies two methods, based on DES and SHA-1, for pseudorandomly generating private keys x and per-message secrets k.