CS8792

CRYPTOGRAPHY AND NETWORK SECURITY

UNIT 3 NOTES

UNIT III PUBLIC KEY CRYPTOGRAPHY

MATHEMATICS OF ASYMMETRIC KEY CRYPTOGRAPHY: Primes – Primality Testing – Factorization – Euler‘s totient function, Fermat‘s and Euler‘s Theorem – Chinese Remainder Theorem – Exponentiation and logarithm – ASYMMETRIC KEY CIPHERS: RSA cryptosystem – Key distribution – Key management – Diffie Hellman key exchange –Elgamal Cryptosystem -- Elliptic curve arithmetic-Elliptic curve cryptography.

3.1 PRIMES

An integer $p > 1$ is a prime number if and only if its only divisors[2] are ± 1 and $\pm p$. **Prime numbers** play a critical role in number theory and in the techniques discussed in this chapter. Table 8.1 shows the primes less than 2000. Note the way the primes are distributed. In particular, note the number of primes in each range of 100 numbers.

Any integer $a > 1$ can be factored in a unique way as

$$a = p_1^{a_1} \times p_2^{a_2} \times \cdots \times p_t^{a_t} \quad \textbf{(8.1)}$$

where $p_1 < p_2 < \ldots < p_t$ are prime numbers and where each a_i is a positive integer. This is known as the fundamental theorem of arithmetic; a proof can be found in any text on number theory.

$$\begin{aligned} 91 &= 7 \times 13 \\ 3600 &= 2^4 \times 3^2 \times 5^2 \\ 11011 &= 7 \times 11^2 \times 13 \end{aligned}$$

It is useful for what follows to express this another way. If P is the set of all prime numbers, then any positive integer a can be written uniquely in the following form:

$$a = \prod_{p \in P} p^{a_p} \qquad \text{where each } a_p \geq 0$$

The right-hand side is the product over all possible prime numbers p; for any particular value of a, most of the exponents a_p will be 0.

The value of any given positive integer can be specified by simply listing all the nonzero exponents in the foregoing formulation.

The integer 12 is represented by $\{a_2 = 2, a_3 = 1\}$.
The integer 18 is represented by $\{a_2 = 1, a_3 = 2\}$.
The integer 91 is represented by $\{a_7 = 1, a_{13} = 1\}$.

Multiplication of two numbers is equivalent to adding the corresponding exponents. Given $a = \prod_{p \in P} p^{a_p}$, $b = \prod_{p \in P} p^{b_p}$. Define $k = ab$. We know that the integer

k can be expressed as the product of powers of primes: $k = \prod_{p \in P} p^{k_p}$. It follows that $k_p = a_p + b_p$ for all $p \in P$.

$$\boxed{\begin{array}{l} k = 12 \times 18 = (2^2 \times 3) \times (2 \times 3^2) = 216 \\ k_2 = 2 + 1 = 3;\ k_3 = 1 + 2 = 3 \\ 216 = 2^3 \times 3^3 = 8 \times 27 \end{array}}$$

What does it mean, in terms of the prime factors of a and b, to say that a divides b? Any integer of the form p^n can be divided only by an integer that is of a lesser or equal power of the same prime number, p^j with $j \leq n$. Thus, we can say the following.

Given

$$a = \prod_{p \in P} p^{a_p},\ b = \prod_{p \in P} p^{b_p}$$

If a|b, then $a_p \leq b_p$ for all p.

$$\boxed{\begin{array}{l} a \ = 12;\ b = 36;\ 12|36 \\ 12 = 2^2 \times 3;\ 36 = 2^2 \times 3^2 \\ a_2 \ = 2 = b_2 \\ a_3 \ = 1 \leq 2 = b_3 \\ \text{Thus, the inequality } a_p \leq b_p \text{ is satisfied for all prime numbers.} \end{array}}$$

It is easy to determine the greatest common divisor[3] of two positive integers if we express each integer as the product of primes.

$$\boxed{\begin{array}{r} 300 = 2^2 \times 3^1 \times 5^2 \\ 18 = 2^1 \times 3^2 \\ \gcd(18, 300) = 2^1 \times 3^1 \times 5^0 = 6 \end{array}}$$

The following relationship always holds:

If $k = \gcd(a, b)$, then $k_p = \min(a_p, b_p)$ for all p.

Determining the prime factors of a large number is no easy task, so the preceding relationship does not directly lead to a practical method of calculating the greatest common divisor.

3.2 EULER'S TOTIENT FUNCTION, FERMAT'S AND EULER'S THEOREM

Two theorems that play important roles in public-key cryptography are Fermat's theorem and Euler's theorem.

Fermat's Theorem[4]

Fermat's theorem states the following: If p is prime and a is a positive integer not divisible by p, then

$$a^{p-1} \equiv 1 \pmod{p} \qquad \textbf{(8.2)}$$

Proof: Consider the set of positive integers less than p: $\{1, 2, \ldots, p-1\}$ and multiply each element by a, modulo p, to get the set $X = \{a \bmod p, 2a \bmod p, \ldots, (p-1)a \bmod p\}$. None of the elements of X is equal to zero because p does not divide a. Furthermore, no two of the integers in X are equal. To see this, assume that $ja \equiv ka \pmod{p})$, where $1 \leq j < k \leq p - 1$. Because a is relatively prime[5] to p, we can eliminate a from both sides of the equation [see Equation (4.3)] resulting in $j \equiv k \pmod{p}$. This last equality is impossible, because j and k are both positive integers less than p. Therefore, we know that the $(p - 1)$ elements of X are all positive integers with no two elements equal. We can conclude the X consists of the set of integers $\{1, 2, \ldots, p - 1\}$ in some order. Multiplying the numbers in both sets (p and X) and taking the result mod p yields

$$a \times 2a \times \ldots \times (p-1)a \equiv [(1 \times 2 \times \ldots \times (p-1)] \pmod{p}$$
$$a^{p-1}(p-1)! \equiv (p-1)! \pmod{p}$$

We can cancel the $((p - 1)!$ term because it is relatively prime to p [see Equation (4.5)]. This yields Equation (8.2), which completes the proof.

$a = 7, p = 19$
$7^2 = 49 \equiv 11 \pmod{19}$
$7^4 \equiv 121 \equiv 7 \pmod{19}$
$7^8 \equiv 49 \equiv 11 \pmod{19}$
$7^{16} \equiv 121 \equiv 7 \pmod{19}$
$a^{p-1} = 7^{18} = 7^{16} \times 7^2 \equiv 7 \times 11 \equiv 1 \pmod{19}$

An alternative form of Fermat's theorem is also useful: If p is prime and a is a positive integer, then

$$a^p \equiv a \pmod{p} \qquad \textbf{(8.3)}$$

Note that the first form of the theorem [Equation (8.2)] requires that a be relatively prime to p, but this form does not.

$p = 5, a = 3 \qquad a^p = 3^5 = 243 \equiv 3(\text{mod}\,5) = a(\text{mod}\,p)$
$p = 5, a = 10 \qquad a^p = 10^5 = 100000 \equiv 10(\text{mod}\,5) \equiv 0(\text{mod}\,5) = a(\text{mod}\,p)$

Euler's Totient Function

Before presenting Euler's theorem, we need to introduce an important quantity in number theory, referred to as **Euler's totient function**, written $\phi(n)$, and defined as the number of positive integers less than n and relatively prime to n. By convention, $\phi(1) = 1$.

DETERMINE $\phi(37)$ AND $\phi(35)$.

Because 37 is prime, all of the positive integers from 1 through 36 are relatively prime to 37. Thus $\phi(37) = 36$.

To determine $\phi(35)$, we list all of the positive integers less than 35 that are relatively prime to it:

1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18
19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34

There are 24 numbers on the list, so $\phi(35) = 24$.

Table 8.2 lists the first 30 values of $\phi(n)$. The value $\phi(1)$ is without meaning but is defined to have the value 1.

Table 8.2 lists the first 30 values of $\phi(n)$. The value $\phi(1)$ is without meaning but is defined to have the value 1.

It should be clear that, for a prime number p,

$$\phi(p) = p - 1$$

Now suppose that we have two prime numbers p and q with $p \neq q$. Then we can show that, for $n = pq$,

$$\phi(n) = \phi(pq) = \phi(p) \times \phi(q) = (p - 1) \times (q - 1)$$

To see that $\phi(n) = \phi(p) \times \phi(q)$, consider that the set of positive integers less that n is the set $\{1, \ldots, (pq - 1)\}$. The integers in this set that are not relatively prime to n are the set $\{p, 2p, \ldots, (q - 1)p\}$ and the set $\{q, 2q, \ldots, (p - 1)q\}$. Accordingly,

$$\begin{aligned}\phi(n) &= (pq - 1) - [(q - 1) + (p - 1)] \\ &= pq - (p + q) + 1 \\ &= (p - 1) \times (q - 1) \\ &= \phi(p) \times \phi(q)\end{aligned}$$

Table 8.2 Some Values of Euler's Totient Function $\phi(n)$

n	$\phi(n)$
1	1
2	1
3	2
4	2
5	4
6	2
7	6
8	4
9	6
10	4
11	10
12	4
13	12
14	6
15	8
16	8
17	16
18	6
19	18
20	8
21	12
22	10
23	22
24	8
25	20
26	12
27	18
28	12
29	28
30	8

$\phi(21) = \phi(3) \times \phi(7) = (3 - 1) \times (7 - 1) = 2 \times 6 = 12$
where the 12 integers are $\{1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20\}$.

Euler's Theorem

Euler's theorem states that for every a and n that are relatively prime:

$$a^{\phi(n)} \equiv 1(\bmod n) \tag{8.4}$$

Proof: Equation (8.4) is true if n is prime, because in that case, $\phi(n) = (n - 1)$ and Fermat's theorem holds. However, it also holds for any integer n. Recall that $\phi(n)$ is the number of positive integers less than n that are relatively prime to n. Consider the set of such integers, labeled as

$$R = \{x_1, x_2, \ldots, x_{\phi(n)}\}$$

That is, each element x_i of R is a unique positive integer less than n with $\gcd(x_i, n) = 1$. Now multiply each element by a, modulo n:

$$S = \{(ax_1 \bmod n), (ax_2 \bmod n), \ldots, (ax_{\phi(n)} \bmod n)\}$$

The set S is a permutation[6] of R, by the following line of reasoning:

1. Because a is relatively prime to n and x_i is relatively prime to n, ax_i must also be relatively prime to n. Thus, all the members of S are integers that are less than n and that are relatively prime to n.

2. There are no duplicates in S. Refer to Equation (4.5). If $ax_i \bmod n = ax_j \bmod n$, then $x_i = x_j$.

Therefore,

$$\prod_{i=1}^{\phi(n)}(ax_i \bmod n) = \prod_{i=1}^{\phi(n)} x_i$$

$$\prod_{i=1}^{\phi(n)} ax_i \equiv \prod_{i=1}^{\phi(n)} x_i (\text{mod } n)$$

$$a^{\phi(n)} \times \left[\prod_{i=1}^{\phi(n)} x_i\right] \equiv \prod_{i=1}^{\phi(n)} x_i (\text{mod } n)$$

$$a^{\phi(n)} \equiv 1 (\text{mod } n)$$

which completes the proof. This is the same line of reasoning applied to the proof of Fermat's theorem.

$a = 3; n = 10;\ \phi(10) = 4 \quad a^{\phi(n)} = 3^4 = 81 = 1 (\text{mod } 10) = 1 (\text{mod } n)$
$a = 2; n = 11;\ \phi(11) = 10 \quad a^{\phi(n)} = 2^{10} = 1024 = 1 (\text{mod } 11) = 1 (\text{mod } n)$

As is the case for Fermat's theorem, an alternative form of the theorem is also useful:

$$a^{\phi(n)+1} \equiv a (\text{mod } n) \qquad \textbf{(8.5)}$$

Again, similar to the case with Fermat's theorem, the first form of Euler's theorem [Equation (8.4)] requires that a be relatively prime to n, but this form does not.

3.3 CHINESE REMAINDER THEOREM

One of the most useful results of number theory is the **Chinese remainder theorem** (CRT).[8] In essence, the CRT says it is possible to reconstruct integers in a certain range from their residues modulo a set of pairwise relatively prime moduli.

The CRT can be stated in several ways. We present here a formulation that is most useful from the point of view of this text. An alternative formulation is explored in Problem 8.17. Let

$$M = \prod_{i=1}^{k} m_i$$

where the m_i are pairwise relatively prime; that is, $\gcd(m_i, m_j) = 1$ for $1 \leq i, j \leq k$, and $i \neq j$. We can represent any integer A in Z_M by a k-tuple whose elements are in Z_{m_i} using the following correspondence:

$$A \leftrightarrow (a_1, a_2, \ldots, a_k) \qquad \textbf{(8.7)}$$

where $A \in Z_M$, $a_i \in Z_{m_i}$, and $a_i = A \bmod m_i$ for $1 \leq i \leq k$. The CRT makes two assertions.

1. The mapping of Equation (8.7) is a one-to-one correspondence (called a **bijection**) between Z_M and the Cartesian product $Z_{m_1} \times Z_{m_2} \times \ldots \times Z_{m_k}$. That is, for every integer A such that $0 \leq A \leq M$, there is a unique k-tuple $(a_1, a_2, \ldots, a_k)$ with $0 \leq a_i < m_i$ that represents it, and for every such k-tuple $(a_1, a_2, \ldots, a_k)$, there is a unique integer A in Z_M.
2. Operations performed on the elements of Z_M can be equivalently performed on the corresponding k-tuples by performing the operation independently in each coordinate position in the appropriate system.

Let us demonstrate the **first assertion**. The transformation from A to $(a_1, a_2, \ldots, a_k)$, is obviously unique; that is, each a_i is uniquely calculated as $a_i = A \bmod m_i$. Computing A from $(a_1, a_2, \ldots, a_k)$ can be done as follows. Let $M_i = M/m_i$ for $1 \leq i \leq k$. Note that $M_i = m_1 \times m_2 \times \ldots \times m_{i-1} \times m_{i+1} \times \ldots \times m_k$, so that $M_i \equiv 0 \pmod{m_j}$ for all $j \neq i$. Then let

$$c_i = M_i \times \left(M_i^{-1} \bmod m_i\right) \qquad \text{for } 1 \leq i \leq k \qquad \textbf{(8.8)}$$

By the definition of M_i, it is relatively prime to m_i and therefore has a unique multiplicative inverse mod m_i. So Equation (8.8) is well defined and produces a unique value c_i. We can now compute

$$A \equiv \left(\sum_{i-1}^{k} a_i c_i\right) (\bmod M) \qquad \textbf{(8.9)}$$

To show that the value of A produced by Equation (8.9) is correct, we must show that $a_i = A \bmod m_i$ for $1 \leq i \leq k$. Note that $c_j \equiv M_j \equiv 0 \pmod{m_i}$ if $j \neq i$, and that $c_i \equiv 1 \pmod{m_i}$. It follows that $a_i = A \bmod m_i$.

The **second assertion** of the CRT, concerning arithmetic operations, follows from the rules for modular arithmetic. That is, the second assertion can be stated as follows: If

$$A \leftrightarrow (a_1, a_2, \ldots, a_k)$$
$$B \leftrightarrow (b_1, b_2, \ldots, b_k)$$

then

$$(A + B) \bmod M \leftrightarrow ((a_1 + b_1) \bmod m_1, \ldots, (a_k + b_k) \bmod m_k)$$
$$(A - B) \bmod M \leftrightarrow ((a_1 - b_1) \bmod m_1, \ldots, (a_k - b_k) \bmod m_k)$$
$$(A \times B) \bmod M \leftrightarrow ((a_1 \times b_1) \bmod m_1, \ldots, (a_k \times b_k) \bmod m_k)$$

3.4 PRIMALITY TESTING

Miller-Rabin Algorithm[7]

The algorithm due to Miller and Rabin [MILL75, RABI80] is typically used to test a large number for primality. Before explaining the algorithm, we need some background. First, any positive odd integer $n \geq 3$ can be expressed as

$$n - 1 = 2^k q \qquad \text{with } k > 0, q \text{ odd}$$

To see this, note that $n - 1$ is an even integer. Then, divide $(n - 1)$ by 2 until the result is an odd number q, for a total of k divisions. If n is expressed as a binary number, then the result is achieved by shifting the number to the right until the rightmost digit is a 1, for a total of k shifts. We now develop two properties of prime numbers that we will need.

Two Properties of Prime Numbers The **first property** is stated as follows: If p is prime and a is a positive integer less than p, then $a^2 \bmod p = 1$ if and only if either $a \bmod p = 1$ or $a \bmod p = -1 \bmod p = p - 1$. By the rules of modular arithmetic $(a \bmod p)(a \bmod p) = a^2 \bmod p$. Thus, if either $a \bmod p = 1$ or $a \bmod p = -1$, then $a^2 \bmod p = 1$. Conversely, if $a^2 \bmod p = 1$, then $(a \bmod p)^2 = 1$, which is true only for $a \bmod p = 1$ or $a \bmod p = -1$

The **second property** is stated as follows: Let p be a prime number greater than 2. We can then write $p - 1 = 2^k q$ with $k > 0$, q odd. Let a be any integer in the range $1 < a < p - 1$. Then one of the two following conditions is true.

1. a^q is congruent to 1 modulo p. That is, $a^q \bmod p = 1$, or equivalently, $a^q \equiv 1 (\bmod\, p)$.
2. One of the numbers $a^q, a^{2q}, a^{4q}, \ldots, a^{2^{k-1}q}$ is congruent to -1 modulo p. That is, there is some number j in the range $(1 \le j \le k)$ such that $a^{2^{j-1}q} \bmod p = -1 \bmod p = p - 1$ or equivalently, $a^{2^{j-1}q} \equiv -1 (\bmod\, p)$.

Proof: Fermat's theorem [Equation (8.2)] states that $a^{n-1} \equiv 1 (\bmod\, n)$ if n is prime. We have $p - 1 = 2^k q$. Thus, we know that $a^{p-1} \bmod p = a^{2^k q} \bmod p = 1$. Thus, if we look at the sequence of numbers

$$a^q \bmod p,\ a^{2q} \bmod p,\ a^{4q} \bmod p, \ldots, a^{2^{k-1}q} \bmod p,\ a^{2^k q} \bmod p \qquad \textbf{(8.6)}$$

we know that the last number in the list has value 1. Further, each number in the list is the square of the previous number. Therefore, one of the following possibilities must be true.

1. The first number on the list, and therefore all subsequent numbers on the list, equals 1.
2. Some number on the list does not equal 1, but its square mod p does equal 1. By virtue of the first property of prime numbers defined above, we know that the only number that satisfies this condition is $p - 1$. So, in this case, the list contains an element equal to $p - 1$.

This completes the proof.

DETAILS OF THE ALGORITHM These considerations lead to the conclusion that, if n is prime, then either the first element in the list of residues, or remainders, $(a^q, a^{2q}, \ldots, a^{2^{k-1}q}, a^{2^k q})$ modulo n equals 1; or some element in the list equals $(n - 1)$; otherwise n is composite (i.e., not a prime). On the other hand, if the condition is met, that does not necessarily mean that n is prime. For example, if $n = 2047 = 23 \times 89$, then $n - 1 = 2 \times 1023$. We compute $2^{1023} \bmod 2047 = 1$, so that 2047 meets the condition but is not prime.

We can use the preceding property to devise a test for primality. The procedure TEST takes a candidate integer n as input and returns the result **composite** if n is definitely not a prime, and the result **inconclusive** if n may or may not be a prime.

```
TEST ($n$)
1. Find integers $k$, $q$, with $k > 0$, $q$ odd, so that
   ($n - 1 = 2^k q$);
2. Select a random integer $a$, $1 < a < n - 1$;
3. if $a^q \bmod n = 1$ then return("inconclusive");
4. for $j = 0$ to $k - 1$ do
5. if $a^{2^j q} \bmod n = n - 1$ then return("inconclusive");
6. return("composite");
```

3.5 DISCRETE LOGARITHMS

Discrete logarithms are fundamental to a number of public-key algorithms, including Diffie-Hellman key exchange and the digital signature algorithm (DSA). This section provides a brief overview of discrete logarithms. For the interested reader, more detailed developments of this topic can be found in [ORE67] and [LEVE90].

The Powers of an Integer, Modulo *n*

Recall from Euler's theorem [Equation (8.4)] that, for every a and n that are relatively prime,

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

where $\phi(n)$, Euler's totient function, is the number of positive integers less than n and relatively prime to n. Now consider the more general expression:

$$a^m \equiv 1 \pmod{n} \tag{8.10}$$

If a and n are relatively prime, then there is at least one integer m that satisfies Equation (8.10), namely, $M = \phi(n)$. The least positive exponent m for which Equation (8.10) holds is referred to in several ways:

- The order of $a \pmod{n}$
- The exponent to which a belongs $\pmod{n}$
- The length of the period generated by a

More generally, we can say that the highest possible exponent to which a number can belong $\pmod{n}$ is $\phi(n)$. If a number is of this order, it is referred to as a **primitive root** of n. The importance of this notion is that if a is a primitive root of n, then its powers

$$a, a^2, \ldots, a^{\phi(n)}$$

are distinct $\pmod{n}$ and are all relatively prime to n. In particular, for a prime number p, if a is a primitive root of p, then

$$a, a^2, \ldots, a^{p-1}$$

are distinct $\pmod{p}$. For the prime number 19, its primitive roots are 2, 3, 10, 13, 14, and 15.

Not all integers have primitive roots. In fact, the only integers with primitive roots are those of the form 2, 4, p^α, and $2p^\alpha$, where p is any odd prime and α is a positive integer. The proof is not simple but can be found in many number theory books, including [ORE76].

Logarithms for Modular Arithmetic

With ordinary positive real numbers, the logarithm function is the inverse of exponentiation. An analogous function exists for modular arithmetic.

Let us briefly review the properties of ordinary logarithms. The logarithm of a number is defined to be the power to which some positive base (except 1) must be raised in order to equal the number. That is, for base x and for a value y,

$$y = x^{\log_x(y)}$$

The properties of logarithms include

$$\log_x(1) = 0$$

$$\log_x(x) = 1$$

$$\log_x(yz) = log_x(y) + \log_x(z) \tag{8.11}$$

$$\log_x(y^r) = r \times \log_x(y) \tag{8.12}$$

Consider a primitive root a for some prime number p (the argument can be developed for nonprimes as well). Then we know that the powers of a from 1 through $(p - 1)$ produce each integer from 1 through $(p - 1)$ exactly once. We also know that any integer b satisfies

$$b \equiv r \pmod{p} \quad \text{for some } r\text{, where } 0 \leq r \leq (p - 1)$$

by the definition of modular arithmetic. It follows that for any integer b and a primitive root a of prime number p, we can find a unique exponent i such that

$$b \equiv a^i \pmod{p} \quad \text{where } 0 \leq i \leq (p - 1)$$

This exponent i is referred to as the **discrete logarithm** of the number b for the base a (mod p). We denote this value as $\text{dlog}_{a,p}(b)$.[10]

Note the following:

$$\text{dlog}_{a,p}(1) = 0 \quad \text{because } a^0 \bmod p = 1 \bmod p = 1 \tag{8.13}$$

$$\text{dlog}_{a,p}(a) = 1 \quad \text{because } a^1 \bmod p = a \tag{8.14}$$

Here is an example using a nonprime modulus, $n = 9$. Here $\phi(n) = 6$ and $a = 2$ is a primitive root. We compute the various powers of a and find

$$\begin{array}{ll} 2^0 = 1 & 2^4 \equiv 7 \pmod 9 \\ 2^1 = 2 & 2^5 \equiv 5 \pmod 9 \\ 2^2 = 4 & 2^6 \equiv 1 \pmod 9 \\ 2^3 = 8 & \end{array}$$

This gives us the following table of the numbers with given discrete logarithms (mod 9) for the root $a = 2$:

Logarithm	0	1	2	3	4	5
Number	1	2	4	8	7	5

To make it easy to obtain the discrete logarithms of a given number, we rearrange the table:

Number	1	2	4	5	7	8
Logarithm	0	1	2	5	4	3

Now consider

$$x = a^{\mathrm{dlog}_{a,p}(x)} \bmod p \qquad y = a^{\mathrm{dlog}_{a,p}(y)} \bmod p$$
$$xy = a^{\mathrm{dlog}_{a,p}(xy)} \bmod p$$

Using the rules of modular multiplication,

$$\begin{aligned} xy \bmod p &= [(x \bmod p)(y \bmod p)] \bmod p \\ a^{\mathrm{dlog}_{a,p}(xy)} \bmod p &= \left[\left(a^{\mathrm{dlog}_{a,p}(x)} \bmod p\right)\left(a^{\mathrm{dlog}_{a,p}(y)} \bmod p\right)\right] \bmod p \\ &= \left(a^{\mathrm{dlog}_{a,p}(x)+\mathrm{dlog}_{a,p}(y)}\right) \bmod p \end{aligned}$$

But now consider Euler's theorem, which states that, for every a and n that are relatively prime,

$$a^{\phi(n)} \equiv 1 \pmod n$$

Any positive integer z can be expressed in the form $z = q + k\phi(n)$, with $0 \leq q < \phi(n)$. Therefore, by Euler's theorem,

$$a^z \equiv a^q (\text{mod } n) \qquad \text{if } z \equiv q \text{ mod } \phi(n)$$

Applying this to the foregoing equality, we have

$$\text{dlog}_{a,p}(xy) \equiv [\text{dlog}_{a,p}(x) + \text{dlog}_{a,p}(y)](\text{mod}\phi(p))$$

and generalizing,

$$\text{dlog}_{a,p}(y^r) \equiv [r \times \text{dlog}_{a,p}(y)](\text{mod } \phi(p))$$

This demonstrates the analogy between true logarithms and discrete logarithms.

Keep in mind that unique discrete logarithms mod m to some base a exist only if a is a primitive root of m.

Table 8.4 Tables of Discrete Logarithms, Modulo 19

(a) Discrete logarithms to the base 2, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$log_{2,19}(a)$	18	1	13	2	16	14	6	3	8	17	12	15	5	7	11	4	10	9

(b) Discrete logarithms to the base 3, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$log_{3,19}(a)$	18	7	1	14	4	8	6	3	2	11	12	15	17	13	5	10	16	9

(c) Discrete logarithms to the base 10, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$log_{10,19}(a)$	18	17	5	16	2	4	12	15	10	1	6	3	13	11	7	14	8	9

(d) Discrete logarithms to the base 13, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$log_{13,19}(a)$	18	11	17	4	14	10	12	15	16	7	6	3	1	5	13	8	2	9

(e) Discrete logarithms to the base 14, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$log_{14,19}(a)$	18	13	7	8	10	2	6	3	14	5	12	15	11	1	17	16	4	9

(f) Discrete logarithms to the base 15, modulo 19

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$log_{15,19}(a)$	18	5	11	10	8	16	12	15	4	13	6	3	7	17	1	2	14	9

ASYMMETRIC KEY CIPHERS

3.6 RSA CRYPTOSYSTEM

- The best known and widely regarded as most practical public-key scheme was proposed byRivest, Shamir & Adleman in 1977
- It is a public-key scheme which may be used for encrypting messages, exchanging keys, andcreating digital signatures
- RSA is a public key encryption algorithm based on exponentiation using modular arithmeticto use the scheme, first generate keys:
- Key-Generation by each user consists of:
    - selecting two large primes at random (~100 digit), p, q
    - calculating the system modulus R=p.q p, q pri0mes
    - selecting at random the encryption key e, e < R, gcd(e, F(R)) = 1
    - solving the congruence to find the decryption key d, e.d [[equivalence]] 1 mod [[phi]](R) 0 <= d <= R
    - publishing the public encryption key: K1={e,R}
    - securing the private decryption key: K2={d,p,q}
- Encryption of a message M to obtain ciphertext C is:

C = Me mod R 0 <= d <= R

- Decryption of a ciphertext C to recover the message M is:

M = Cd = Me.d = M1+n.[[phi]](R) = M mod R

The RSA Algorithm

Key Generation

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p * q$	
Calcuate $f(n) = (p - 1)(q - 1)$	
Select integer e	$\gcd(ø(n), e) = 1; 1<e<ø(n)$
Calculate d	$d \equiv e^{-1} \pmod{ø(n)}$
Public key	PU = {e, n}
Private key	PR = {d, n}

Encryption

Plaintext:	$M<n$
Ciphertext:	$C = M^e \bmod n$

Decryption

Ciphertext:	C
Plaintext:	M $=C^d$ mod n

Example:

P=17 q=11 e=7 M=88

1. Select two prime numbers, p = 17 and q = 11.
2. Calculate n = pq = 17 * 11 = 187.
3. Calculate ø(n) = (p - 1)(q - 1) = 16 * 10 = 160.
4. Select e such that e is relatively prime to ø(n) = 160 and less than ø(n); we choose e = 7.
5. Determine d such that de ≡ 1 (mod 160) and d <160. The correct value isd = 23, because 23 * 7 = 161 = (1 * 160) + 1; d can be calculated using the extended Euclid's algorithm.
6. The resulting keys are public key PU={7,187} and private key PR={23,187}.
7. Plaintext input of M = 88.

For encryption,

calculate C = 88^7 mod 187.

88^7 mod 187 = [(88^4 mod 187) * (88^2 mod 187) * (88^1 mod 187)] mod 187

88^1 mod 187 = 88

88^2 mod 187 = 7744 mod 187 = 77

88^4 mod 187 = 59,969,536 mod 187 = 132

88^7 mod 187 = (88 * 77 * 132) mod 187 = 894,432 mod 187 = 11

For decryption,

calculate M = 11^{23} mod 187

11^{23} mod 187 = [(11^1 mod 187) * (11^2 mod 187) * (11^4 mod 187) *
(11^8 mod 187) * (11^8 mod 187)] mod 187

11^1 mod 187 = 11

11^2 mod 187 = 121

11^4 mod 187 = 14,641 mod 187 = 55

11^8 mod 187 = 214,358,881 mod 187 = 33

11^{23} mod 187=(11* 121 * 55 * 33 * 33) mod 187 =79,720,245 mod 187= 88

The Security of RSA

Brute force: This involves trying all possible private keys.

Mathematical attacks: There are several approaches, all equivalent in effort to factoring the product oftwo primes.

Timing attacks: These depend on the running time of the decryption algorithm.

Hardware fault-based attack: This involves inducing hardware faults in the processor that is generatingdigital signatures.

Chosen ciphertext attacks: This type of attack exploits properties of the RSA algorithm.

Solve:

a. p = 3; q = 11, e = 7; M = 5

b. p = 5; q = 11, e = 3; M = 9

c. p = 7; q = 11, e = 17; M = 8

3.7 KEY MANAGEMET

- For symmetric encryption to work, the two parties to an exchange must share the same key, and that key must be protected from access by others.
- For two parties A and B, key distribution can be achieved in a number of ways, as follows:
  1. A can select a key and physically deliver it to B.
  2. A third party can select the key and physically deliver it to A and B.
  3. If A and B have previously and recently used a key, one party can transmit the new key to the other,encrypted using the old key.
  4. If A and B each has an encrypted connection to a third party C, C can deliver a key on the encryptedlinks to A and B.

A Key Distribution Scenario

- User A wishes to establish a logical connection with B and requires a one-time session key to protect the data transmitted over the connection. A has a master key, K_a, known only to itself and the KDC; similarly, B shares the master key K_b with the KDC. The following steps occur.

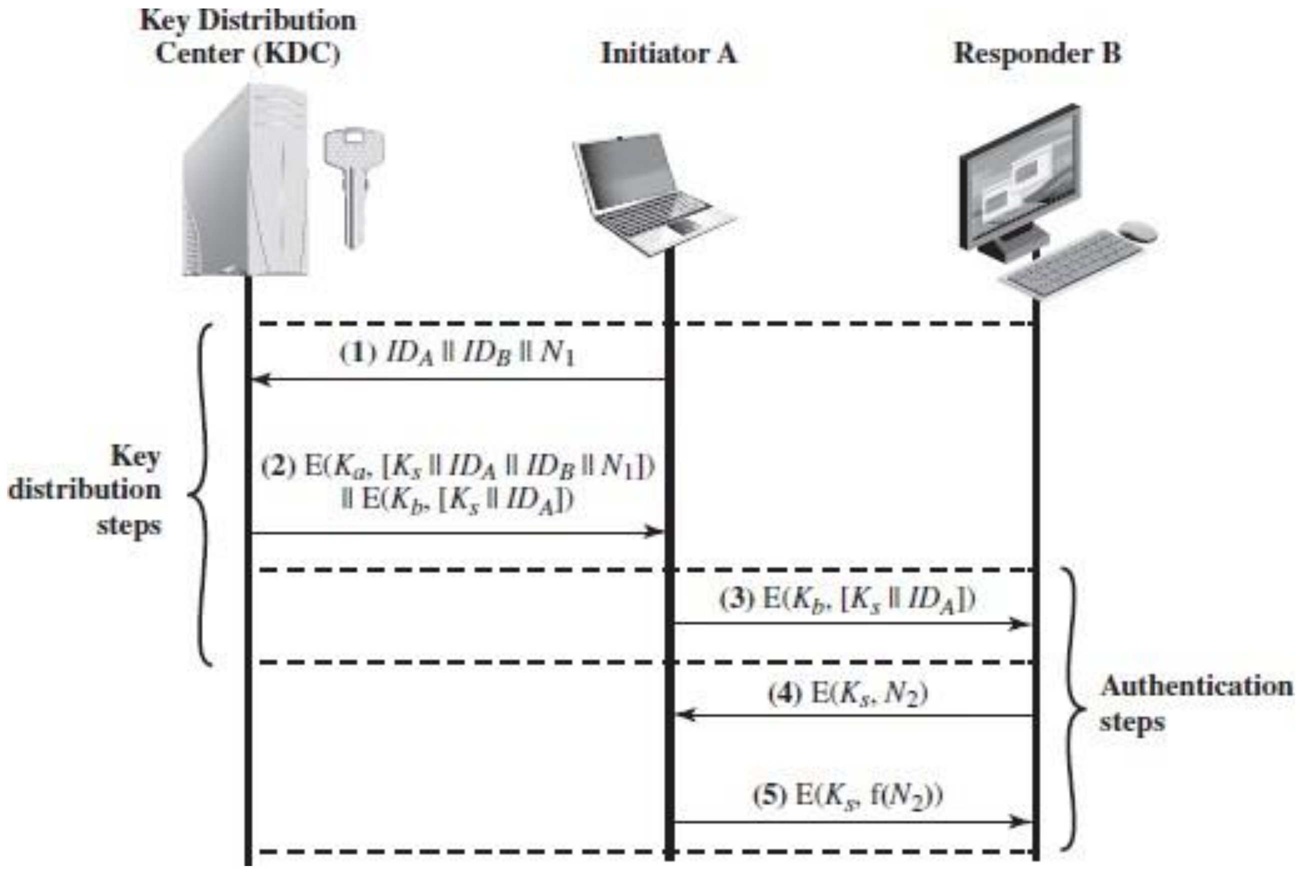
1. A issues a request to the KDC for a session key to protect a logical connection to B. The message includes the identity of A and B and a unique identifier, N_1, for this transaction, which we refer to as a **nonce**. The nonce may be a timestamp, a counter, or a random number; the minimum requirement is that it differs with each request. Also, to prevent masquerade, it should be difficult for an opponent to guess the nonce. Thus, a random

number is a good choice for a nonce.

2. The KDC responds with a message encrypted using K_a. Thus, A is the only one who can successfully read the message, and A knows that it originated at the KDC. The message includes two items intended forA:
   - ➢ The one-time session key, K_s, to be used for the session
   - ➢ The original request message, including the nonce, to enable A to match this response with the appropriate request Thus, A can verify that its original request was not altered before reception by the KDC and, because of the nonce, that this is not a replay of some previous request.
   - ➢ In addition, the message includes two items intended for B:
     - The one-time session key, K_s, to be used for the session
     - An identifier of A (e.g., its network address), ID_A

➢ These last two items are encrypted with K_b (the master key that the KDC shares with B). They are to be sent to B to establish the connection and prove A's identity.

3. A stores the session key for use in the upcoming session and forwards to B the information that originatedat the KDC for B, namely, $E(K_b,[K_s \| ID_A])$.Because this information is encrypted with K_b, it is protected from eavesdropping. B now knows the sessionkey (K_s), knows that the other party is A (from ID_A), and knows that the information originated at the KDC(because it is encrypted using K_b).
4. Using the newly minted session key for encryption, B sends a nonce, N_2, to A.
5. Also, using Ks, A responds with $f(N_2)$, where f is a function
   that performssome transformation on N_2



Key Distribution Center (KDC)
Initiator A
Responder B
(1) $ID_A \| ID_B \| N_1$
Key distribution steps
(2) $E(K_a, [K_s \| ID_A \| ID_B \| N_1]) \| E(K_b, [K_s \| ID_A])$
(3) $E(K_b, [K_s \| ID_A])$
(4) $E(K_s, N_2)$
Authentication steps
(5) $E(K_s, f(N_2))$

Hierarchical Key Control

- It is not necessary to limit the key distribution function to a single KDC.
- A hierarchical scheme minimizes the effort involved in master key distribution, because most master keysare those shared by a local KDC with its local entities.

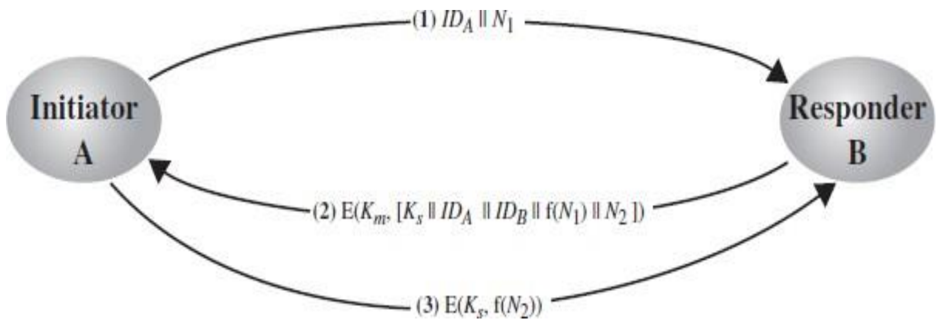
Session Key Lifetime

- The more frequently session keys are exchanged, the more secure they are, because the opponent has less ciphertext to work with for any given session key. On the other hand, the distribution of session keys delays the start of any exchange and places a burden on network capacity. A security manager must try to balance these competing considerations in determining the lifetime of a particular session key.
- For connection-oriented protocols, one obvious choice is to use the same session key for the length of time that the connection is open, using a new session key for each new session. If a logical connection has a verylong lifetime, then it would be prudent to change the session key periodically, perhaps every time the PDU (protocol data unit) sequence number cycles.

- For a connectionless protocol, such as a transaction-oriented protocol, there is no explicit connection initiation or termination. Thus, it is not obvious how often one needs to change the session key. The most secure approach is to use a new session key for each exchange. However, this negates one of the principal benefits of connectionless protocols, which is minimum overhead and delay for each transaction. A better strategy is to use a given session key for a certain fixed period only or for a certain number of transactions.

Decentralized Key Control

- The use of a key distribution center imposes the requirement that the KDC be trusted and be protected from subversion. This requirement can be avoided if key distribution is fully decentralized.
- A decentralized approach requires that each end system be able to communicate in a secure manner with all potential partner end systems for purposes of session key distribution. Thus, there may need to be as many as [n(n - 1)]/2 master keys for a configuration with n end systems.



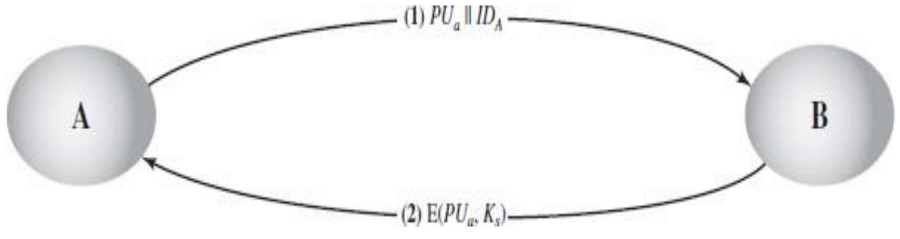
A session key may be established with the following sequence of steps

1. A issues a request to B for a session key and includes a nonce, N1.
2. B responds with a message that is encrypted using the shared master key. The response includes the session key selected by B, an identifier of B, the value f(N1), and another nonce, N2.
3. Using the new session key, A returns f(N2) to B.

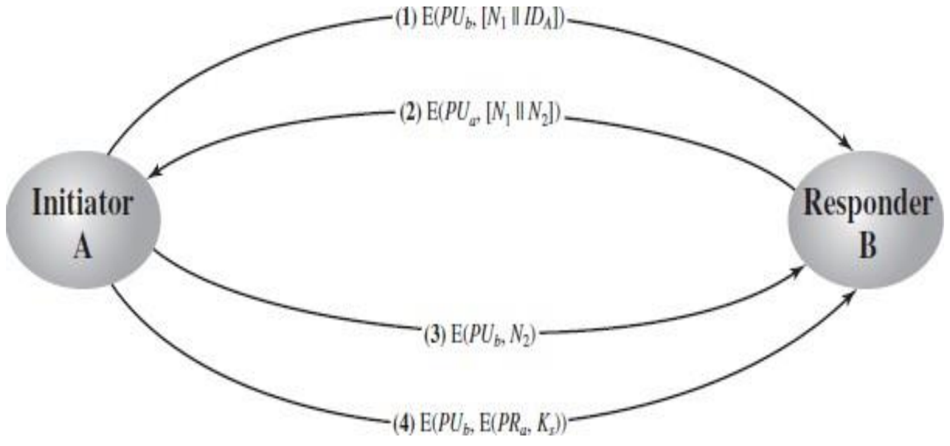
Symmetric Key Distribution using Asymmetric Encryption

Simple Secret Key Distribution

1. A generates a public/private key pair {PUa, PRa} and transmits a message to B consisting of PUa and anidentifier of A, IDA.
2. B generates a secret key, Ks, and transmits it to A, which is encrypted with A's public key.
3. A computes D(PRa, E(PUa, Ks)) to recover the secret key. Because only A can decrypt the message, only A and B will know the identity of Ks.
4. A discards PUa and PRa and B discards PUa.



Public-Key Distribution of Secret Keys



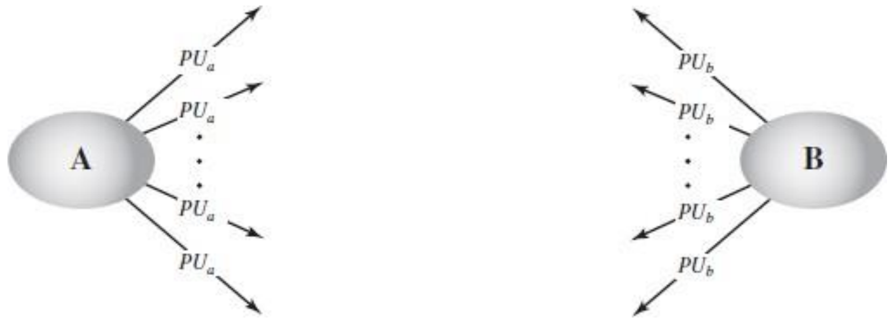
(1) $E(PU_b, [N_1 \| ID_A])$
(2) $E(PU_a, [N_1 \| N_2])$
Initiator A
Responder B
(3) $E(PU_b, N_2)$
(4) $E(PU_b, E(PR_a, K_s))$

1. A uses B's public key to encrypt a message to B containing an identifier of A(IDA) and a nonce (N1), which is used to identify this transaction uniquely.
2. B sends a message to A encrypted with PUa and containing A's nonce (N1) as well as a new nonce generated by B (N2). Because only B could have decrypted message (1), the presence of N1 in message (2)assures A that the correspondent is B.
3. A returns N2, encrypted using B's public key, to assure B that its correspondent is A.
4. A selects a secret key Ks and sends M = E(PUb, E(PRa, Ks)) to B. Encryption of this message with B's public key ensures that only B can read it; encryption with A's private key ensures that only A could havesent it.
5. B computes D(PUa, D(PRb, M)) to recover the secret key.

3.8 KEY DISTRIBUTION

- Public announcement
- Publicly available directory
- Public-key authority
- Public-key certificates

Public Announcement of Public Keys



Publicly Available Directory

1. The authority maintains a directory with a {name, public key} entry for each participant.
2. Each participant registers a public key with the directory authority.

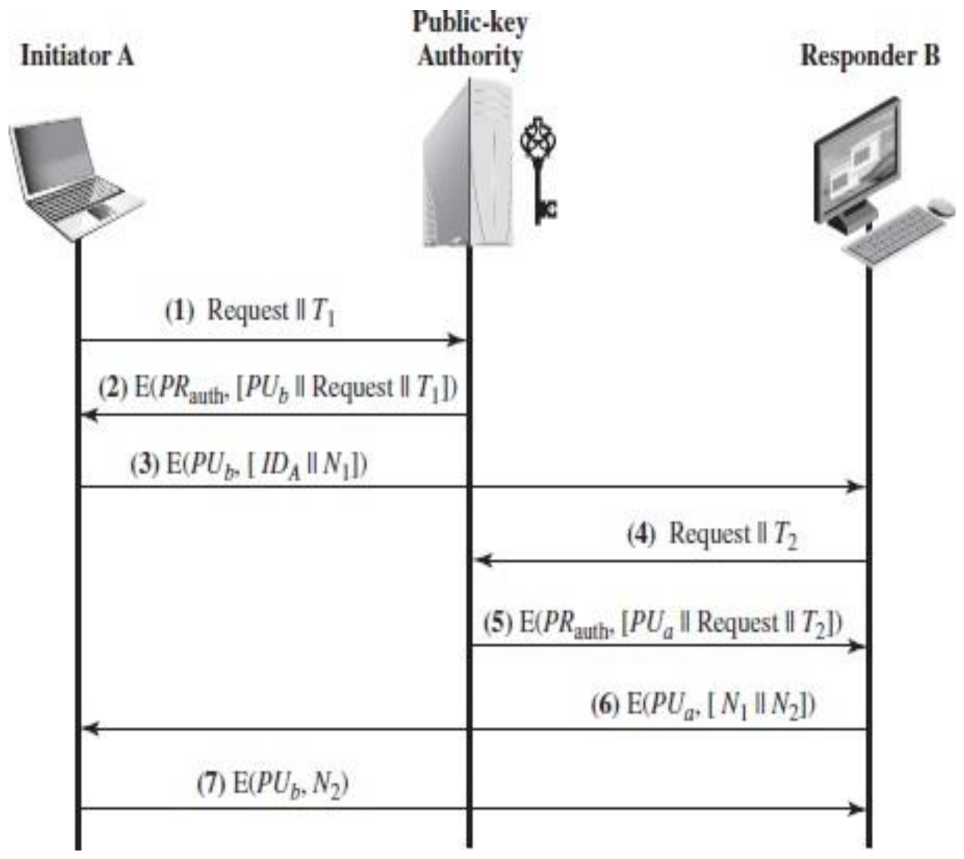
Registration would have to be in person or by some form of secure authenticated communication.

3. A participant may replace the existing key with a new one at any time, either because of the desire to replace a public key that has already been used for a large amount of data, or because the corresponding private key has been compromised in some way.
4. Participants could also access the directory electronically. For this purpose, secure, authenticated communication from the authority to the participant is mandatory.

Public-Key Authority

1. A sends a timestamped message to the public-key authority containing aRequest for the current public key of B.
2. The authority responds with a message that is encrypted using the authority's private key, PRauth. Thus, A is able to decrypt the message using the authority's public key. Therefore, A is assured that the messageoriginated with the authority. The message includes the following:

- B's public key, PUb, which A can use to encrypt messages destined for B
- The original request used to enable A to match this response with the corresponding earlier request and toverify that the original request was not altered before reception by the authority
- The original timestamp given so A can determine that this is not an old message from the authority containing a key other than B's current public key.


Initiator A
Public-key Authority
Responder B
(1) Request ‖ T_1
(2) E(PR_{auth}, [PU_b ‖ Request ‖ T_1])
(3) E(PU_b, [ID_A ‖ N_1])
(4) Request ‖ T_2
(5) E(PR_{auth}, [PU_a ‖ Request ‖ T_2])
(6) E(PU_a, [N_1 ‖ N_2])
(7) E(PU_b, N_2)

3. A stores B's public key and also uses it to encrypt a message to B containing an identifier of A (ID_A) anda nonce (N_1), which is used to identify this transaction uniquely.

4, 5. B retrieves A's public key from the authority in the same manner as A Retrieved B's public key.

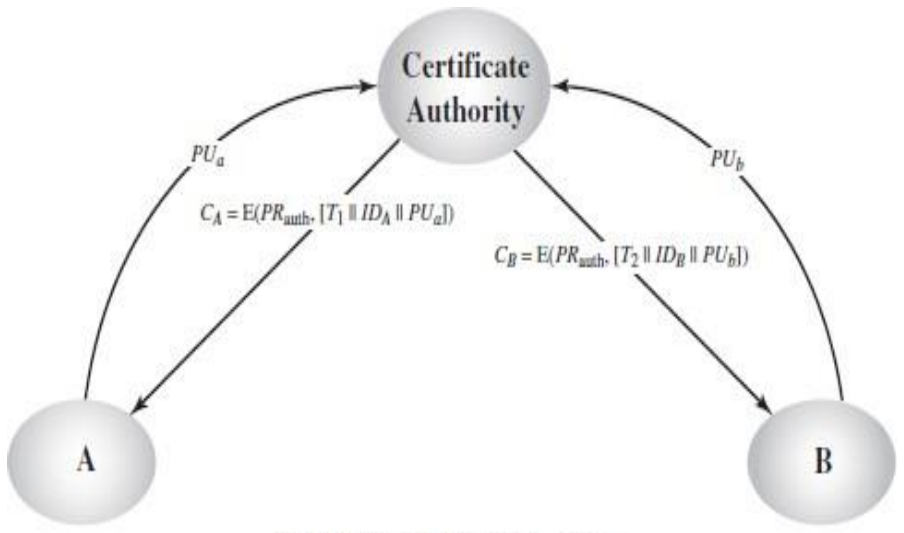
At this point, public keys have been securely delivered to A and B, and they may begin their protected exchange. However, two additional steps are desirable:

6. B sends a message to A encrypted with PU_a and containing A's nonce (N_1) as well as a new nonce generated by B (N_2). Because only B could have decrypted message (3), the presence of N_1 in message (6)assures A that the correspondent is B.

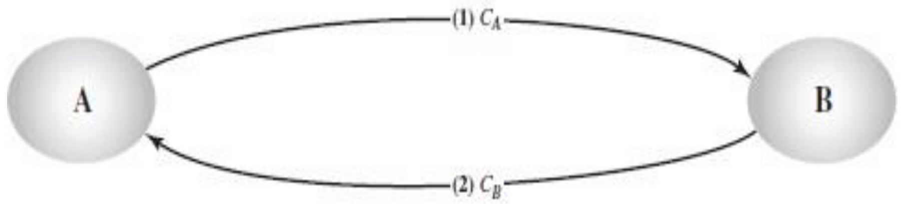
7. A returns N_2, which is encrypted using B's public key, to assure B that its Correspondent is A.

Public-Key Certificates

1. Any participant can read a certificate to determine the name and public key of the certificate’s owner.
2. Any participant can verify that the certificate originated from the certificate authority and is notcounterfeit.
3. Only the certificate authority can create and update certificates.
4. Any participant can verify the currency of the certificate.


Certificate Authority
PU_a
PU_b
$C_A = E(PR_{auth}, [T_1 \| ID_A \| PU_a])$
$C_B = E(PR_{auth}, [T_2 \| ID_B \| PU_b])$
A
B

(a) Obtaining certificates from CA



(b) Exchanging certificates

For participant A, the authority provides a certificate of the form$C_A = E(PR_{auth}, [T \| ID_A \| PU_a])$
where PR_{auth} is the private key used by the authority and T is a timestamp. $D(PU_{auth}, C_A) = D(PU_{auth}, E(PR_{auth}, [T \| ID_A \|PU_a])) = (T \| ID_A \|PU_a)$

3.8 DIFFIE HELLMAN KEY EXCHANGE

- The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for subsequent encryption of messages. The algorithm itself is limited to the exchange of secret values.
- The Diffie-Hellman algorithm depends for its effectiveness on the difficulty of computing discretelogarithms.

Diffie Hellman key exchange as follows:

1. There are publicly known numbers: a prime number „q‟and an integer α that is primitive root of q.
2. suppose users A and B wish to exchange a key. User A selects a random integer XA< q and computes YA = α^{XA} mod q.
3. Similarly, user B independently selects a random integer XB < q and computes YB = α^{XB} mod q.
4. Each side keeps the X value private and makes the Y value available publicly to the other side
5. User A computes the key as K = $(YB)^{XA}$ mod q and User B computes the key as K = $(YA)^{XB}$ mod q
6. These two calculations produce identical results.

$$\begin{aligned} K &= (YB)^{XA} \bmod q \\ &= (\alpha^{XB} \bmod q)^{XA} \bmod q \\ &= (\alpha^{XB})^{XA} \bmod q \\ &= (\alpha^{XA})^{XB} \bmod q \\ &= (\alpha^{XA} \bmod q)^{XB} \bmod q \\ &= (YA)^{XB} \bmod q \end{aligned}$$

The result is that two sides have exchanged a secret key. The security of the algorithm lies in the fact that, while it is relatively easy tocalculate Exponentials modulo a prime, it is very difficult to calculate discrete logarithms. For large primes,the latter task is considered infeasible.

Diffie-Hellman Key Exchange

Global Public Elements

q	prime number
α	α<q, and α is a primitive root of q

User A Key Generation

Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{X_A} \bmod q$

User B Key Generation

Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{X_B} \bmod q$

Calculation of Secret Key by User A

$K = (Y_B)^{X_A} \bmod q$

Calculation of Secret Key by User B

$K = (Y_A)^{X_B} \bmod q$

Example:

Key exchange is based on the use of the prime number
$q = 353$ and a primitive root of 353, in this case $\alpha = 3$. A and B select private keys
$X_A = 97$ and $X_B = 233$, respectively.

A computes $Y_A = 3^{97} \bmod 353 = 40$.

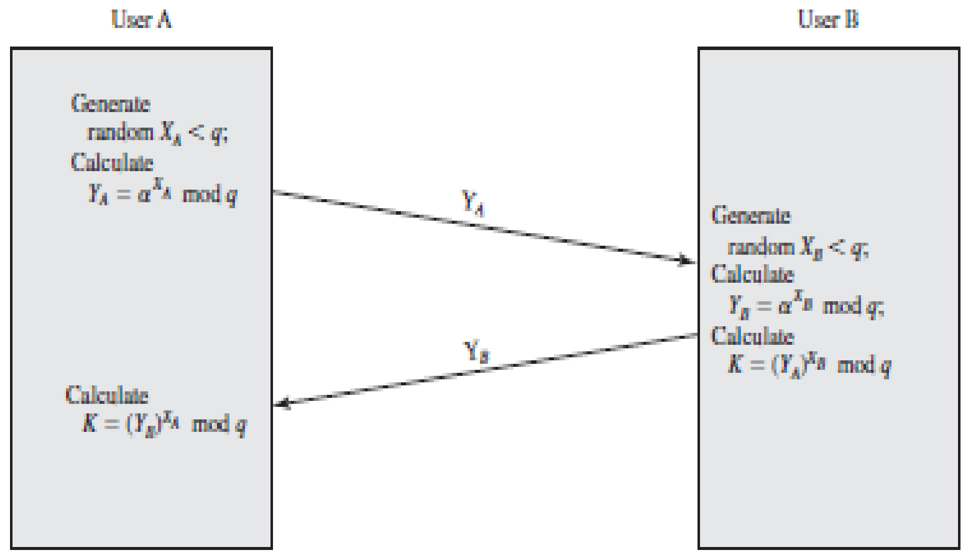
B computes $Y_B = 3^{233} \bmod 353 = 248$.

After they exchange public keys, each can compute the common secret key:

A computes $K = (Y_B)^{X_A} \bmod 353 = 248^{97} \bmod 353 = 160$.

B computes $K = (Y_A)^{X_B} \bmod 353 = 40^{233} \bmod 353 = 160$.

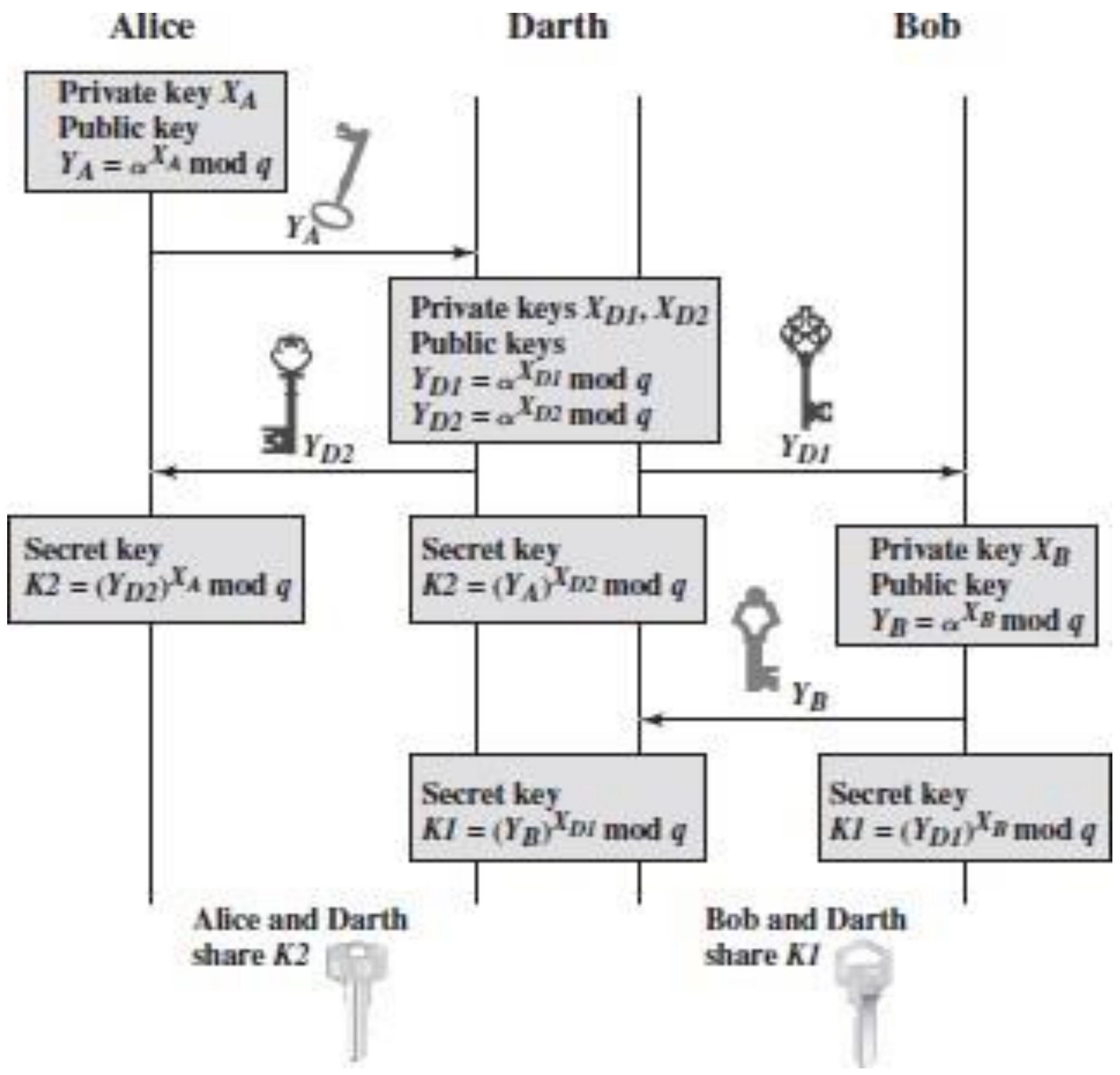
Key Exchange Protocols


User A
User B
Generate random $X_A < q$;
Calculate $Y_A = \alpha^{X_A} \bmod q$
Y_A
Generate random $X_B < q$;
Calculate $Y_B = \alpha^{X_B} \bmod q$;
Calculate $K = (Y_A)^{X_B} \bmod q$
Y_B
Calculate $K = (Y_B)^{X_A} \bmod q$

Man-in-the-Middle Attack

Suppose Alice and Bob wish to exchange keys, and Darth is the adversary. The attack proceeds asfollows.

1. Darth prepares for the attack by generating two random private keys XD1 and XD2 and then computing the corresponding public keys YD1 and YD2.
2. Alice transmits YA to Bob.
3. Darth intercepts YA and transmits YD1 to Bob. Darth also calculatesK2 = (YA)^{XD2} mod q.
4. Bob receives YD1 and calculates K1 = (YD1)^{XB} mod q.
5. Bob transmits YB to Alice.
6. Darth intercepts YB and transmits YD2 to Alice. Darth calculatesK1 = (YB)^{XD1} mod q.
7. Alice receives YD2 and calculates K2 = (YD2)^{XA} mod q.



Alice
Darth
Bob
Private key X_A
Public key
$Y_A = \alpha^{X_A} \bmod q$
Y_A
Private keys X_{D1}, X_{D2}
Public keys
$Y_{D1} = \alpha^{X_{D1}} \bmod q$
$Y_{D2} = \alpha^{X_{D2}} \bmod q$
Y_{D2}
Y_{D1}
Secret key
$K2 = (Y_{D2})^{X_A} \bmod q$
Secret key
$K2 = (Y_A)^{X_{D2}} \bmod q$
Private key X_B
Public key
$Y_B = \alpha^{X_B} \bmod q$
Y_B
Secret key
$K1 = (Y_B)^{X_{D1}} \bmod q$
Secret key
$K1 = (Y_{D1})^{X_B} \bmod q$
Alice and Darth share $K2$
Bob and Darth share $K1$

Bob and Darth share secret key K1 and Alice and Darth share secret key K2. All future communicationbetween Bob and Alice is compromised in the following way.

1. Alice sends an encrypted message M: E(K2, M).
2. Darth intercepts the encrypted message and decrypts it to recover M.
3. Darth sends Bob E(K1, M) or E(K1, M′), where M′ is any message. In the first case, Darth simply wants to eavesdrop on the communication without altering it. In the second case, Darth wants to modify the message going to Bob.

3.9 Elgamal Cryptosystem

- The variant of the Diffie-Hellman key distribution scheme, allowing secure exchange of Messages published in 1985 by ElGamal in T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms",
- Like Diffie-Hellman its security depends on the difficulty of factoring logarithms

Key Generation

- select a large prime p (~200 digit), and
- [[alpha]] a primitive element mod p
- A has a secret number xA
- B has a secret number xB
- A and B compute yA and yB respectively, which are then made publicy

A = [[alpha]]xA mod p

B = [[alpha]]xB mod p

to **encrypt** a message **M** into ciphertext **C**,

- selects a random number **k,** 0 <= k <= p-1
- computes the message key **K**

K = yB k mod p

- computes the ciphertext pair: C = {c1,c2}

C1 = [[alpha]]k mod p C2 = K.M mod pto **decrypt** the message

-
- extracts the message key **K**

K = C1 xB mod p = [[alpha]]k.xB mod p

- extracts **M** by solving for M in the following equation:C2 = K.M mod p

Algorithm

Global Public Elements

q	prime number
α	$\alpha<q$, and α is a primitive root of q
Key Generation by Alice	
Select private X_A	$X_A < q - 1$
Calculate Y_A	$Y_A = \alpha^{X_A} \bmod q$
Public key	$\{q, a, Y_A\}$
Private key	X_A

Encryption by Bob with Alice's Public Key

Plaintext:	$M < q$
Select random integer k	$k < q$
Calculate K	$K = (Y_A)k \bmod q$
Calculate C_1	$C_1 = \alpha^k \bmod q$
Calculate C_2	$C_2 = KM \bmod q$
Ciphertext:	(C_1, C_2)

Decryption by Alice with Alice's Private Key

Ciphertext:	(C_1, C_2)
Calculate K	$K = (C_1)^{X_A} \bmod q$
Plaintext:	$M = (C_2K^{-1}) \bmod q$

1. Bob generates a random integer k.
2. Bob generates a one-time key K using Alice's public-key components Y_A, q, and k.
3. Bob encrypts k using the public-key component a, yielding C_1. C_1 provides sufficient information for Alice to recover K.
4. Bob encrypts the plaintext message M using K.
5. Alice recovers K from C1 using her private key.
6. Alice uses K^{-1} to recover the plaintext message from C_2.

Thus, K functions as a one-time key, used to encrypt and decrypt the message.
Example:
let us start with the prime field GF(19); that is, $q = 19$.
It has primitive roots {2, 3, 10, 13, 14, 15}. choose $\alpha = 10$.

1. Alice chooses $X_A = 5$.
2. Then $Y_A = \alpha^{X_A} \bmod q = \alpha^5 \bmod 19 = 3$.
3. Alice's private key is 5 and Alice's public key is $\{q, a, Y_A\} = \{19, 10, 3\}$.

Suppose Bob wants to send the message with the value M = 17. Then,

1. Bob chooses k = 6.
2. Then $K = (Y_A)^k \bmod q = 3^6 \bmod 19 = 729 \bmod 19 = 7$.
3. $C_1 = \alpha^k \bmod q = \alpha^6 \bmod 19 = 11$
   $C_2 = KM \bmod q = 7 * 17 \bmod 19 = 119 \bmod 19 = 5$
4. Bob sends the ciphertext (11, 5).

For decryption:
1. Alice calculates $K = (C_1)^{X_A} \bmod q = 11^5 \bmod 19 = 161051 \bmod 19 = 7$.
2. Then K^{-1} in GF(19) is $7^{-1} \bmod 19 = 11$.
3. Finally, $M = (C_2K^{-1}) \bmod q = 5 * 11 \bmod 19 = 55 \bmod 19 = 17$.

If a message must be broken up into blocks and sent as a sequence of encrypted blocks, a unique value of k should be used for each block. If k is used for more than one block, knowledge of one block M1 of the message enables the user to compute other blocks as follows.
Let
$C_{1,1} = \alpha^k \bmod q$; $C_{2,1} = KM_1 \bmod q$ $C_{1,2} = \alpha^k \bmod q$; $C_{2,2} = KM_2 \bmod q$ Then,
$C_{2,1}/C_{2,2} = (KM_1 \bmod q / KM_2 \bmod q) = (M_1 \bmod q / M_2 \bmod q)$ If M_1 is known, then M_2 is easily computed as
$M_2 = (C_{2,1})^{-1} C_{2,2} M_1 \bmod q$

3.10Elliptic Curve Arithmetic and Elliptic Curve Cryptography (ECC)

The principal attraction of ECC, compared to RSA, is that it appears to offer equal security for a far smaller key size, thereby reducing processing overhead.

Abelian Group:

An abelian group G, sometimes denoted by {G, • }, is a set of elements with a binary operation, denoted by •, that associates to each ordered pair (a, b) of elements in G an element (a • b) in G

(A1) Closure: If a and b belong to G, then a • b is also in G.

(A2) Associative: a • (b • c) = (a • b) • c for all a, b, c in G.

(A3) Identity element: There is an element e in G such that a • e = e • a = afor all a in G.

(A4) Inverse element: For each a in G there is an element a′ in G such thata • a′ = a′ • a = e.

(A5) Commutative: a•b = b•a for all a, b in G.

Elliptic Curves over Real Numbers

Elliptic curves are not ellipses. They are so named because they are described by cubic equations, similar to those used for calculating the circumference of an ellipse. In general, cubic equations for elliptic curvestake the following form, known as a **Weierstrass equation:**

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

where a, b, c, d, e are real numbers and x and y take on values in the real numbers.

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

Such equations are said to be cubic, or of degree 3, because the highest exponent they contain is a 3. Also included in the definition of an elliptic curve is a single element denoted O and called the point at infinity or the zero point

$$y = \sqrt{x^3+ax+b}$$

1. O serves as the additive identity. Thus O = -O; for any point P on the elliptic curve, P + O = P. In whatfollows, we assume P ≠ O and Q ≠ O.

2. The negative of a point P is the point with the same x coordinate but the negative of the y coordinate;that is, if P = (x, y), then -P = (x, -y).

Note that these two points can be joined by a vertical line.Note that P + (-P) = P - P = O.

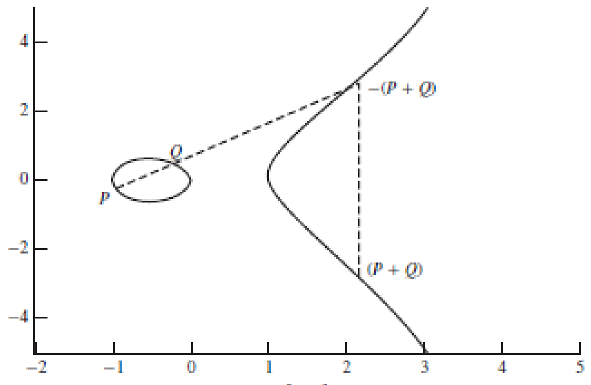
3. To add two points P and Q with different x coordinates, draw a straight line between them and find thethird point of intersection R. It is easily seen that there is a unique point R that is the point of intersection

(unless the line is tangent to the curve at either P or Q, in which case we take R = P or R = Q, respectively).To form a group structure, we need to define addition on these three points:

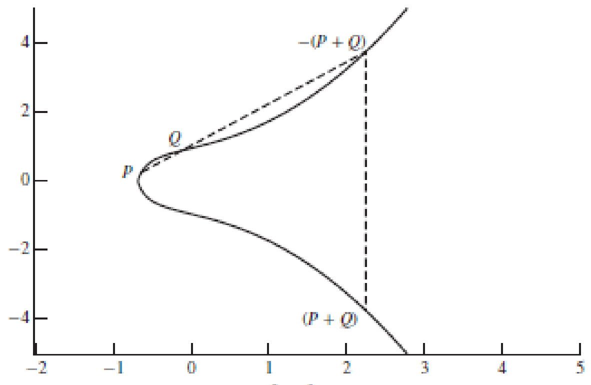
P + Q = -R. That is, we define P + Q to be the mirror image (with respect to the x axis) of the third point of intersection.

4. The geometric interpretation of the preceding item also applies to two points, P and -P, with the same x coordinate. The points are joined by a vertical line, which can be viewed as also intersecting the curve at the infinity point. We therefore have P + (-P) = O, which is consistent with item (2).

5. To double a point Q, draw the tangent line and find the other point of intersection S. Then Q + Q = 2Q =-S.



(a) $y^2 = x^3 - x$



(b) $y^2 = x^3 + x + 1$

Algebraic Description of Addition

P = (x_P, y_P) and Q = (x_Q, y_Q), that are not negatives of each other, the slope of the line l that joins them is $\Delta = (y_Q - y_P)/(x_Q - x_P)$. There is exactly one other point where l intersects the elliptic curve, and that is thenegative of the sum of P and Q.

After some algebraic manipulation, we can express the sum

R = P + Q as$x_R = \Delta^2 - x_P - x_Q$

$y_R = -y_P + \Delta(x_P - x_R)$

P + P = 2P = R. When $y_P \neq 0$

$$x_R = \left(\frac{3x_P^2 + a}{2y_P}\right)^2 - 2x_P$$

$$y_R = \left(\frac{3x_P^2 + a}{2y_P}\right)(x_P - x_R) - y_P$$

ECC Diffie-Hellman Key ExchangeGlobal Public Elements

$E_q(a, b)$	elliptic curve with parameters *a*, *b*, and *q*, where *q* is a prime or an integer of the form 2^m
G	point on elliptic curve whose order is largevalue *n*

User A Key Generation

Select private n_A	$n_A < n$
Calculate public P_A	$P_A = n_A * G$

User B Key Generation

Select private n_B	$n_B < n$
Calculate public P_B	$P_B = n_B * G$

Calculation of Secret Key by User A

$K = n_A * P_B$

Calculation of Secret Key by User B

$K = n_B * P_A$