Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Prime Numbers in Public Key Cryptography

The subject of prime numbers has fascinated mathematicians for centuries. Some of the methods for finding prime numbers date to antiquity. The properties of primes have been investigated for thousands of years. The advent of digital computers and public-key cryptography has brought the subject of prime numbers into the mainstream and focused renewed attention on it. Prime numbers are interesting entities (you may not believe that now but I'm confident that you will when you have finished reading this paper) that are ex...

Copyright SANS Institute
Author Retains Full Rights

AD

Prime Numbers in Public Key Cryptography

An Introduction

Jerry Crow

ABSTRACT

The use of public-key cryptography is pervasive in the information protection and privacy arenas. Public key crypto algorithms utilize prime numbers extensively; indeed, prime numbers are an essential part of the major public key systems. This paper provides an introduction to prime numbers and how they are chosen, identified and used in public key systems. The content of this paper is specifically targeted at an audience that has only basic mathematical knowledge. A reader who has taken a high school or college algebra class should be able to follow the math herein.

The objective of this paper is to inform the mainstream information security professional – who does not necessarily possess an extensive knowledge of mathematics – about the nature of prime numbers and how they are used in contemporary public key systems, thereby increasing his/her overall understanding of contemporary asymmetric encryption algorithms. As part of this investigation the basic elements of Diffie-Hellman exchange and the RSA algorithm are explored.

GSEC Practical Assignment Version 1.4b

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Table of Contents

1. OVERVIEW 1

2. DEFINITIONS 1

3. PROPERTIES OF PRIMES 2

3.1. PRIME PAIRS 2
3.2. MERSENNE PRIMES 3
3.3. PERFECT NUMBERS 4
3.4. THE GOLDBACH CONJECTURE 4

4. MODULAR ARITHMETIC 5

4.1. DEFINITIONS 5
4.2. OPERATIONS 5

5. IMPORTANT THEOREMS 6

5.1. FERMAT'S LITTLE THEOREM 6
5.2. EULER'S TOTIENT FUNCTION 6
5.3. EULER'S THEOREM 6

6. TESTING FOR PRIMALITY 7

6.1. FERMAT'S LITTLE THEOREM 7
6.2. SIEVE OF ERATOSTHENES 7
6.3. PRIMES IS IN P 7

7. DIFFIE-HELLMAN EXCHANGE 8

8. RSA 10

9. ALGORITHM SECURITY 12

10. SUGGESTED FURTHER READING 13

REFERENCES 14

© SANS Institute 2003, Author retains full rights

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

1. OVERVIEW

The subject of prime numbers has fascinated mathematicians for centuries. Some of the methods for finding prime numbers date to antiquity. The properties of primes have been investigated for thousands of years. The advent of digital computers and public-key cryptography has brought the subject of prime numbers into the mainstream and focused renewed attention on it.

Prime numbers are interesting entities (you may not believe that now but I'm confident that you will when you have finished reading this paper) that are extremely important in many branches of mathematics. They are also essential to a number of real world algorithms including most of the algorithms used in public key (asymmetric) cryptography.

This paper explores some of the basic properties of prime numbers and several theorems associated with them. It also presents moderate detail on two of the most common asymmetric algorithms and the manner in which they employ prime numbers.

2. DEFINITIONS

We will begin by defining some basic terms. If you have some depth in mathematics you may skip to section 3 without loss of continuity. Most college algebra textbooks such as Miller, et al. [5] cover this material in greater detail and more formally. There is also more detail in Menezes, et al. [11].

Integer: the mathematics presented herein deals with integers. An integer is a whole number without a fractional part. 1, 2, 3, … etc. We will deal only with positive integers; i.e., no negative numbers. We say that an integer n divides an integer m if m/n (where n is not zero) has a remainder of zero. This property is commonly referred to as m being "evenly divisible" by n. More formally this property is expressed as:

$$m/n = f \qquad (m>n,\ n \neq 0)$$

Factor: In this relationship n is called a factor of m (f is also a factor of m). For example, 8 is a factor of 24; 3 is a factor of 9; 5 is not a factor of 21 because 21 is not evenly divisible (i.e., remainder = 0) by 5.

Prime number: Every integer is evenly divisible by itself and 1. A prime number is an integer that is <u>only</u> divisible by itself and 1. Examples: 11, 13, 17, 19, 23 … A number that is not prime is called composite. Euclid (yes, *the* Euclid of plane geometry [among other things] fame) proved in about 300 BCE that the number of prime numbers is infinite. As integer values get quite large, however, the distance between primes gets larger; i.e., the number of primes in any linear group of ascending order integers becomes smaller. The primes less than 100 are shown below

```
2 3 5 7 11 13 17 19 21 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```


© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

All decimal prime numbers >5 end in (have a least significant digit of) 1, 3, 7 or 9 because no even number greater than 2 is prime and any decimal number ending in 5 is evenly divisible by 5. As numeric values get larger, primes get rarer. Only about four percent of the first 25 billion integers are prime. This uneven, unpredictable distribution of primes among the integers contributes to the difficulty of locating candidates for large prime numbers and determining if a chosen candidate is prime.

Relatively Prime Numbers: two integers are called relatively prime to one another if they have no common factors other than 1. The numbers themselves need not be prime. In formal notation this is expressed as

$$\gcd(M, N) = 1$$

i.e., the greatest common denominator (largest common factor) of the two numbers is 1. For example, 8 and 15 are relatively prime because they have no common factors other than 1. 12 and 15 are not relatively prime because they share the common factor 3. Relatively prime numbers are important in asymmetric cryptography; it is important to understand the difference between prime numbers and relatively prime numbers and understand that two numbers that are not prime (e.g., 8 and 15) may still be relatively prime.

All prime numbers are by definition relatively prime to one another. Several of the algorithms presented below require the selection of a pair of relatively prime numbers. The simplest way to ensure that two numbers are relatively prime is to select a pair of primes. This is particularly true when the numbers are large because determining all the factors of extremely large integers is a non-trivial exercise (a fact that, as we shall see, is a foundation of the strength of the algorithms presented below).

3. PROPERTIES OF PRIMES

This section presents some of the interesting properties of prime numbers. The investigation of the properties of primes can lead to new algorithms for determining if a number is prime or for finding the prime factors of large integers.

3.1. Prime Pairs

As mentioned above there is an infinite number of primes. Many primes occur in pairs on either side of an even integer. For example, 11|13, 17|19. There is probably an infinite number of these pairs (sometimes called "prime twins") as well (though this has not been proven), but their frequency gets smaller as the integer values get larger. For example, note there are eight such pairs in the list of primes <100

2 3 5 7 11 13 17 19 21 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

but there are only five such pairs between 100 and 200, only three between 200 and 300 and only two between 300 and 400.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

3.2. Mersenne Primes

Any positive integer that is one less than an integral power of 2 can be expressed as

$$2^n - 1$$

where *n* is a positive integer. Many such numbers are prime. Put another way, it turns out that some integers that are exactly one less than an even power of 2 are prime. The concept of expressing prime numbers in the form 2^n–1 dates to antiquity, but the first major work published about such primes was authored by a French monk named Marin Mersenne (1588-1648) in the 17th century and prime numbers that can be expressed in this form are now called Mersenne primes. Note that not all prime numbers can be expressed in this manner (e.g., 11). For simplicity I will henceforth use M_n to represent the Mersenne prime 2^n – 1. For example, $M_3 = 2^3-1 = 7$, the second Mersenne prime.

The search for ever larger Mersenne primes is an icon of today's recreational mathematical pursuits. In late 2001 the largest Mersenne prime yet discovered was verified by a 20-year-old resident of Ontario, Canada; he was participating in a distributed search managed by software provided by Entropia, Inc. The number has 4,053,946 digits – yes, that's the number of digits in the Mersenne prime expressed in decimal, not the number itself. The Mersenne prime is

$$2^{13,466,917} - 1$$

an incomprehensibly huge number. The number was identified by means of the Great Internet Mersenne Prime Search (GIMPS) (www.mersenne.org) [14].

Early writings about Mersenne primes conjectured that 2^n-1 is prime for all primes n; i.e., it was believed that one less than 2^n is prime for every case where n is prime. In 1536, however, it was shown that M_{11} is composite.

$$M_{11} = 2^{11}-1 = 2047 = 23 * 89$$

A number of later writings contained incorrect assertions about which M_n values are prime, including those of Mersenne himself. Mersenne asserted in 1644 that M_n is prime for

```
n = 2, 3, 5, 7, 13, 17, 19, 67, 127, 257
```

and that all other M_n are composite for n < 257. This is incorrect (the values above 19 are actually 31, 61, 89, 107 and 127; the next is 521), though the errors were not discovered until hundreds of years later (Euler added 31 to the list in 1750). There are many interesting theorems related to Mersenne numbers and primes. For example, if M_n is prime then n is also prime (though, as noted above, the converse is not necessarily true). These theorems are very useful in modern number theory and factoring investigations. Such investigations might one day yield an efficient algorithm for factoring very large numbers into their prime factors, a breakthrough (or not, depending upon how one looks at it) that would render most contemporary public key cryptography systems useless.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

3.3. Perfect Numbers

A *perfect number* is an integer > 1 which meets the condition that the sum of its positive divisors exclusive of the number itself is equal to the number (some sources say "equal to twice the number" and include the number itself; the definitions are equivalent). For example, 6 is a perfect number because the sum of 1, 2, 3 (the divisors of 6) is 6. The next perfect number is 28 = 1+2+4+7+14. The next two perfect numbers are 496 and 8128. These first four perfect numbers were known to the ancient Greeks.

It can be shown that an even integer greater than 1 is perfect if and only if it has the form $2^{n-1}(2^n-1)$ and the latter term, 2^n-1, is prime; i.e., any perfect number is a prime multiplied by some power of 2. Note that the latter term is a Mersenne prime. Relationships such as this one lead to interesting and potentially useful facts such as a perfect number always has a Mersenne prime as a factor. As noted above, facts such as these may one day collectively contribute to the discovery of an algorithm for efficient (i.e., fast) factoring of large numbers.

3.4. The Goldbach Conjecture

In 1742, Christian Goldbach, a mathematician and historian, asserted that every even number could be expressed as the sum of two prime numbers. For example, 6 = 3 + 3, 20 = 17 + 3, etc. This assertion, which is widely known as the Goldbach conjecture, is one of a well-known (at least in mathematics circles) group of yet-to-be proved mathematical assertions, conjectures, theorems, etc. The most famous of this group was probably Fermat's Last Theorem which states that there exist no integer solutions for $X^n + Y^n = Z^n$ for n > 2. This theorem (which wasn't first posited by Pierre de Fermat, but the reason it bears his name is outside the scope of this discussion) was formally proved by a Princeton mathematician in the 1990s, and the Goldbach conjecture may well be the most widely known remaining problem. No formal proof of the conjecture has ever been put forth, but the validity of the conjecture has been verified by computation. In 2000 that verification was extended to 4×10^{14} using a network of computers. In other words, the Goldbach conjecture has been tested and found true for every even number from 2 to 400,000,000,000,000. Beyond that, it has been tested for selected even numbers up to 10^{300}.

The Goldbach conjecture is widely assumed to be true for all even numbers because as the magnitude of the numbers increases, more than one pair of primes is found to satisfy the conjecture. For example, there are two pairs for the number 20: 17 + 3 and 13 + 7. For very large even numbers there are dozens of such pairs and for extremely large even numbers there are thousands of such pairs.

Still, the conjecture remains unproven by formal methods. If you would like to win at least a million dollars (a prize offered by a British firm), not to mention gaining a substantial amount of fame, all you need do is prove, using formal

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

mathematical constructs, that every even number can be expressed as the sum of two prime numbers.

4. MODULAR ARITHMETIC

Modular arithmetic is an essential part of the major contemporary asymmetric encryption systems. In fact, as we shall see, the use of modular arithmetic is the key breakthrough that enables both Diffie-Hellman exchange and RSA to work.

4.1. Definitions

Modular arithmetic involves division of integers with an associated remainder. Given two integers, *m* and *n*, *m* mod *n* is defined as the remainder when m is divided by n. This is commonly written as

$$m \bmod n = r$$

Where *r* is the remainder of *m/n*. If *r* = 0, then *n* divides *m* as noted in section 2 above. The term r is sometimes referred to as the *residue*. Examples

$$4 \bmod 3 = 1; \quad 21 \bmod 5 = 1; \quad 20 \bmod 4 = 0$$

If *m* mod *k* = *n* mod *k* the two integers *m* and *n* are said to be congruent modulo k. This congruence is written

$$m \equiv n \bmod k$$

4.2. Operations

Basic algebra defines the properties of association, commutation and distribution for the operations of addition and multiplication as

Associative property: $(x + y) + z = x + (y + z)$; $xy(z) = x(yz)$

Commutative property: $x + y = y + x$; $xy = yx$

Distributive property: $x(y + z) = xy + xz$

Modular operations exhibit these same properties. For multiplication, our primary interest here, we have

Association: $((w * x) * y) \bmod n = (w * (x * y)) \bmod n$

Commutation: $(w * x) \bmod n = (x * w) \bmod n$

Distribution: $(w * (x * y)) \bmod n = ((w * x) * (w * y)) \bmod n$

Of greater interest are the following relationships:

IF $(x + y) \equiv (x + z) \bmod n$ THEN $y \equiv z \bmod n$

IF $(x * y) \equiv (x * z) \bmod n$ THEN $y \equiv z \bmod n$

The first relationship is true for all positive integers. The latter relationship, however, is true *only* if x is relatively prime to n. This latter relationship is important in modular arithmetic based encryption. More detail will be found in Stallings [1].

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

5. IMPORTANT THEOREMS

Two major theorems and one function are important components of asymmetric cryptography. These are Fermat's theorem (sometimes called Fermat's Little Theorem), named for Pierre de Fermat (1601-1665) and Euler's theorem and Euler's Totient function, named for Leonhard Euler (1707-1783). The names Fermat and Euler appear in any realistic list of the greatest mathematicians of all time.

The theorems presented below can be proved (i.e., shown to be true using formal mathematical constructs), but such proofs are beyond the scope of this paper.

5.1. Fermat's Little Theorem

Fermat's little theorem states that for any prime p and any positive integer m not divisible by p

$$m^{p-1} \equiv 1 \bmod p$$

The theorem is sometimes written in an alternative form, true for any m

$$m^{p} \equiv m \bmod p$$

Expressed in this manner, the theorem means that for any prime p if we divide m^p by p and m by p we will get the same remainder. This relationship is useful for testing numbers to see if they are prime (it can only be used to exclude; i.e., if the remainders of the two operations are not the same, the number is definitely not prime, but if they are it *may* be prime).

5.2. Euler's Totient Function

Euler's totient function (sometimes called Euler's *phi* function), written $F(n)$, is the number of integers less than n and relatively prime to n. For example, $F(10) = 4$ (the numbers 1, 3, 7, 9). The usefulness of the totient function in this discussion is a result of the fact that for any prime p, $F(p) = p - 1$. From this and the operations noted in the previous section it follows that for any two primes p and q and their product n

$$F(n) = F(pq) = F(p) * F(q) = (p-1) * (q-1)$$

We will see this relationship again below.

5.3. Euler's Theorem

Euler's theorem says that for any two integers m and n that are relatively prime the following is true

$$m^{F(n)} \equiv 1 \bmod n$$

If we generalize this congruence as

$$m^{p} \equiv 1 \bmod n$$

we know by virtue of Euler's theorem that there is at least one value of p that satisfies the relation -- $F(p)$; there may be more.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

6. TESTING FOR PRIMALITY

We have shown that prime numbers exhibit certain interesting properties. We have presented some of the more important theorems involving prime numbers. It has been mentioned several times that prime numbers – particularly very large prime numbers – are important in asymmetric cryptography. It follows, therefore, that the task of ensuring that a particular number is prime has taken on practical importance as the use of public-key cryptography has become widespread.

As noted in section 2 the percentage of prime numbers in any group of integers decreases as numbers get larger and larger. Yet the need to isolate primes, particularly very large primes, has become increasingly important because of the importance of such large prime numbers to public-key cryptography algorithms.

There is no known method for rapidly and conclusively testing a given number for primality. Until just recently, the algorithms available, particularly those that could be executed in a reasonable amount of time, could only conclusively exclude a number (prove it composite) or show that a given integer *might* be a prime. Several methods for testing for primes are summarized below. Details on additional, more complex algorithms for primality testing will be found in Menezes, et al. [11] and on the Internet at the URL [15].

6.1. Fermat's Little Theorem

As described above, Fermat's little theorem provides a fast method for excluding a number. For any integer m and a possibly prime number p, if $m^p \bmod p \neq m \bmod p$ then p is not prime. If the remainders are equal, p may be prime. Contemporary usage scenarios of this algorithm are conducted by repeatedly testing p using randomly chosen values of m. It can be shown that each consecutive time the test is valid (the remainders are equal) the probability that p is composite becomes smaller. If this process is repeated a sufficient number of times the probability that p is prime becomes very nearly 100%.

6.2. Sieve of Eratosthenes

A process named for a Greek mathematician of the third century BCE, the algorithm is a trial and error based method for testing a number for primality. If no prime greater than 2 and smaller than the square root of the number being tested divides that number (is a factor of the number) then the number is prime. The sieve is slow and becomes computationally expensive and time consuming as the magnitude of the number being tested increases.

6.3. PRIMES is in P

PRIMES is in P is the somewhat odd title of a paper recently produced by a professor and two of his graduate students in India. The paper [12] describes a method for testing primes that yields a conclusive result. The algorithm is not particularly fast – the authors concede as much – but the paper has been

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

heralded because the algorithm presented will definitively determine if a number is prime and it is applicable to any integer. In addition, the algorithm executes in what is called "polynomial time". Many primality testing algorithms scale in exponential time which means that as the numbers being tested grow larger the time required to test them increases extremely rapidly. The algorithm presented by this paper scales in polynomial time which means that the time required to test increases more slowly as the value of the number being tested increases.

7. DIFFIE-HELLMAN EXCHANGE

Historically, all serious cryptography involved the use of symmetric algorithms. Such schemes specify the use of a single key for both encryption and decryption. The fundamental problem of private-key crypto, when conducted on a large scale, is key creation and distribution. The challenge is formidable and the solution is complex and expensive – so much so that, with few exceptions, only nation states and their military organizations could tolerate the expense of secure key distribution. Thus, given strong algorithms, the effective deployment of symmetric cryptography to protect information exchange comes down to effective key management. The problem of key management has been central to symmetric crypto since the middle ages and was made even more difficult by the advent of wireless communication in the 20th century. As widespread use of radio-based communication emerged the problem of providing the single key, in advance, to both participants in an asymmetrically encrypted information exchange received a lot of attention. An effective solution to this problem became the holy grail of private-key (symmetric) cryptography (the narrative in Kahn [10] contains many references to this problem).

In 1976 Whitfield Diffie and Martin Hellman, at Stanford, published a paper [6] that detailed a method for two parties to securely obtain a private key for subsequent use in a symmetric encryption session without physically exchanging anything and without ever exchanging the private key. Based upon work done by them and Ralph Merkle, the paper is regarded as the initial step toward public-key cryptography and represented the answer to the key distribution problem noted above. In a world where information is transported by networks, Diffie-Hellman key exchange (as it has become known) is a secure method for two parties exchanging information over the Internet (for example) to agree upon a session key for use with a strong symmetric algorithm (e.g., AES, IDEA).

Diffie-Hellman exchange is based upon modular arithmetic; indeed, it was the combined use of exponentiation and modular arithmetic that provided the breakthrough needed to arrive at the scheme. The exchange is based upon a so-called "one-way" function, an arithmetic operation that is easy to do or perform but extremely difficult to undo or reverse. In this case the function is $G^x \bmod P$. Given values for G and P it is straightforward to calculate $G^x \bmod P$. Given the result of this calculation, however, it is extremely difficult to determine the value of x. Technically, the security of the system relies upon the difficulty of calculating discrete logarithms in a finite field, an issue that is usually referred to as the discrete

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

logarithm problem. If you wish to know more detail about this, a web search will return pointers to more information than you probably desire. Schneier [2] and Menezes, et al. [11] also describe this problem.

Let us assume that Alice and Bob wish to exchange some information in a secure manner and that Eve wishes to eavesdrop on their exchange (these characters have emerged as the "standard" players in cryptography discussions). First, Bob and Alice agree upon values for G and P; these values need not be secret and may be exchanged across an insecure channel. In addition, each selects a secret value that will not be transmitted across the insecure channel. There are some restrictions on these numbers: G must be less than P for example.

The exchange goes like this, using g = 5 and p = 11:

1. Alice selects 2 as her secret key (ka)
2. Bob selects 3 as his secret key (kb)
3. Alice computes an intermediate number $Ia = g^{ka} \bmod p = 5^2 \bmod 11 = 3$
4. Bob computes an intermediate number $Ib = g^{kb} \bmod p = 5^3 \bmod 11 = 4$
5. Each sends the intermediate number to the other; this transmission can take place in the clear; i.e., it does not matter if Eve learns these numbers.
6. Alice computes a final number: $K = Ib^{ka} \bmod p = 4^2 \bmod 11 = 5$
7. Bob computes a final number: $K = Ia^{kb} \bmod p = 3^3 \bmod 11 = 5$

This works because

$$K = Ia^{kb} \bmod p = Ib^{ka} \bmod p = g^{(ka*kb)} \bmod p$$

And, as noted, it's strong and secure because deriving K without the knowledge of ka and kb is extremely difficult.

Note that Bob and Alice now have agreed upon a private key – the number 5 – without ever exchanging the key itself. Moreover, because of the difficulty of undoing the modular functions, it is not feasible for Eve to derive this key even if she learns the values of g, p and both of the intermediate numbers. This example, of course, would be easy to reverse. In practice these numbers would be on the order of one hundred digits long (or more).

One of the most useful contemporary applications of Diffie-Hellman exchange is the establishment of a one-time session key to be used to drive a symmetrically encrypted exchange of data between two network-based correspondents. The procedure is straightforward and the key thus created can be used with, for example, a DES session or an IDEA session. One of the chief advantages of symmetric encryption algorithms is that they are generally amenable to rapid computation. Public-key algorithms, on the other hand, are extremely expensive computationally. Accordingly, private key algorithms are almost always used to secure the exchange of large volumes of data. The key for such an exchange is created in real time and exchanged using a modular based algorithm such as Diffie-Hellman or a public-key system such as RSA.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Note, however, that the system described above requires, for practical application, that Alice and Bob be on line at the same time; i.e., real-world use of Diffie-Hellman exchange requires real-time communication between the correspondents. This is at best an inconvenience and at worst a serious limitation. In their paper, Diffie and Hellman alluded to the possibility of devising an encryption system in which two keys were used – one to encrypt and the other to decrypt. It was Diffie that first conceived of the idea of an asymmetric encryption system using different keys. The paper did not provide a working example of an asymmetric system, but just describing the concept of such a system was revolutionary at the time. The paper called such a system "public-key" cryptography and noted the advantages of such a system. Users of such a system, should one be devised, would be freed from the real-time constraint noted above. Correspondents could publish one of the keys in a public forum. Anyone desiring to send an encrypted message to a particular individual could retrieve that individual's public key from the forum and use it to encrypt the message. Only the designated recipient, with knowledge of the other key, could decrypt.

It did not take long for someone to build upon the work of Diffie, Hellman and Merkle to create a working public-key system.

8. RSA

The RSA system is largely synonymous with public-key cryptography in the popular press. Certainly it is the most widely used system. Conceived by Rivest, Shamir and Adleman (hence RSA) while the three were collaborating at the Massachusetts Institute of Technology, the system has withstood the test of time and considerable scrutiny since it first emerged in the literature in the late 1970s [7]. The strength of this system relies upon the difficulty of factoring large numbers – specifically the difficulty associated with finding the specific pair of prime numbers selected to create a large integer called the modulus.

Rivest, et al., extended the work of Diffie and Hellman by devising a workable "one-way" function, an equation involving both exponentiation and modular arithmetic. Consider the following:

$$C = P^e \bmod M$$

Computation of C is straightforward given values for P, e and M. As we have seen, however, computing the modular inverse, i.e., finding e given values for C, P and M is difficult. For example, using modular congruence, find e where

$$C \equiv P^e \bmod M$$

Finding a value for e in the above equation given all the other variables is considered infeasible in any realistic amount of time given sufficiently large values of P and M. Suppose that C is the ASCII value of a byte of ciphertext, P is the plaintext, e is the encryption key and M is a value called the modulus. If the modulus is large and e is large, then breaking the encryption of the system involves

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

determining the value of *e* given only the values of M and C; i.e., finding the modular inverse of *e*. This is an extremely hard problem for which no known method of rapid solution has been published.

Suppose, however, that there exists a value *d* such that

$$P \equiv C^d \bmod M$$

i.e., *d* is the value needed to compute the modular inverse of the encryption value.

We have the following requirements:

1. We need a modulus *M* that is sufficiently large to make outright computation of the modular inverse intractable.
2. We need an encryption exponent (key) *e* and a decryption exponent (key) *d* such that

   $$C \equiv P^e \bmod M \qquad \text{and}$$

   $$P \equiv C^d \bmod M$$

   Where *P* and *C* represent a byte of plaintext and ciphertext, respectively.
3. It must be infeasible to determine the value of *d* given the values of *e* and M and infeasible to determine the value of *e* given the values of *d* and *M*.

The system conceived by Rivest, Shamir and Adleman meets these requirements as follows:

1. Alice selects two extremely large prime numbers *p* and *q*. The numbers should be at least 100 decimal digits – perhaps longer – and must be prime. The best security against factoring is provided by selecting numbers that are the same length.
2. Alice computes the modulus *M* by multiplying *p* and *q*. Thus $M = pq$.
3. Alice selects another integer, *e*, the encryption exponent (key), such that $e < (p-1)(q-1)$ and *e* is relatively prime to $(p-1)(q-1)$. The simplest way to ensure these two conditions are met is to select a prime number that is less than $(p-1)(q-1)$.
4. Alice must now determine a final integer *d*, the decryption exponent (key), such that $d < (p-1)(q-1)$ and $ed \equiv 1 \bmod M$. A common method for determining *d* is called the Extended Euclidean Algorithm. [2] and [11] provide details of this algorithm.
5. Alice now publishes the pair (*M*,*e*) as her public key and retains *d* as her private key. The values of *p* and *q* are discarded and never disclosed.

The strength of the system lies in the difficulty of reversing a function that uses exponentials in modular arithmetic. If Bob encrypts a message using Alice's private key, and Eve intercepts the ciphertext C, she cannot recover the plaintext *P* even though she knows the values of *e* and *M* (they are public information) because to do so involves finding *e* where $C \equiv P^e \bmod M$ and, as noted, this is an extremely difficult task (impossible, for practical purposes, given sufficiently large values of M and e). But Alice has computed the special value *d* based upon the secret values *p* and *q*

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

that she selected and *d* can be used to decrypt. Eve cannot break the encryption without knowing the values of *p* and *q* and to find them she must factor *M* into a specific pair of primes.

This works because Euler's theorem says that, given the manner in which *e* and *d* were selected

$$ed \equiv 1 \bmod F(M) \qquad \text{or}$$

$$d \equiv e^{-1} \bmod F(M)$$

which says that *d* and *e* are inverses mod Φ(M). But, Φ(M) = (p-1)(q-1). So the following relationships hold given our selection of values

$$P = C^d \bmod M \equiv (P^e)^d \bmod M \equiv P^{ed} \bmod M \equiv P \bmod M$$

Based upon Euler's theorm and Fermat's little theorem.

The RSA system has been subjected to extreme testing and analysis and is believed very strong given a large modulus. It is interesting to note, however, that it has never been formally proven that it is impossible to rapidly determine the factors of a large integer. If at some point in the future a mathematician devises an algorithm for rapidly factoring large integers, the RSA system will be rendered useless.

9. ALGORITHM SECURITY

While there are other useful methods for conducting public-key cryptography (see [2] for an excellent summary), the RSA system is by far the most widely used. Given prudent procedures for protecting one's private key and diligent application of obvious ancillary security procedures designed to control access to computing platforms, etc., the security of RSA comes down to the difficulty of factoring the modulus. Current literature that speaks to key strength often talks of key lengths – usually expressed in bits. In the case of RSA, the length of interest is the size of the modulus, not the prime numbers used to create it (though, of course, the two are directly related).

Key size recommendations for public-key algorithms are much larger than those for private-key (symmetric) algorithms. At first glance this seems counter-intuitive. Why does a system like RSA, which seems on the surface to be far more complex and sophisticated than a symmetric block cipher like DES, require, for adequate security, larger key sizes than DES or AES? The answer is straightforward: there is no known "backdoor" for a system like AES or DES. Given adequate key distribution and protection methods, the only useful method to attack such a cipher is brute force; i.e., trying every possible key. Hence, key size choices in this realm are based upon estimates of how long it would take for a brute force attack to succeed using contemporary computing technology. On the other hand, there is a "backdoor" into RSA and it can be opened by factoring the modulus. It's far more difficult to estimate how long an attempt to factor a large number would take using the same computing technology. Hence, key size choices here tend to be larger.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Contemporary pundits are recommending an RSA key of at least 1024 bits; i.e., the value of *M* should be at least a 1024 bit number. This size recommendation can only increase as technology continues to provide access to ever more powerful computers. [18] provides a comprehensive analysis of various factors contributing to effective key size choice. In addition, the authors propose a formula for estimating the increase in minimum key size over time. The requirement increases linearly for symmetric keys and exponentially for asymmetric keys. Their work projects that in 2010 symmetric algorithms will require at least 80-bit keys and asymmetric, RSA-based algorithms will need 1450-bit keys. These numbers seem inadequate measured against contemporary literature (the paper [18] is three years old).

Chapter 11 of Denning's work [9] contains an excellent discussion of the relationship between key size and attack difficulty. Schneier [2] also addresses this subject in chapter 7, which ends by providing the advice to select "keys that are longer than you imagine necessary".

10. SUGGESTED FURTHER READING

For those intrepid readers who would like considerably more detail than is presented herein, I offer the following suggestions:

If you are interested in the general history of cryptography and its evolution, from antiquity to about 1965, Kahn's work [10] is considered the definitive source. The version now in print is the second edition and contains an additional chapter summarizing material that was declassified in the 1990s, long after the first edition was published. Singh's work [4] reads well and contains information on contemporary crypto subjects including public-key systems and quantum based systems.

Detailed information regarding the nature of and mathematics surrounding virtually every modern, useful cryptographic algorithm will be found in Schneier's work [2]. This book is regarded by many as the definitive work on cryptographic algorithms. [11] also contains an enormous amount of mathematical detail and it's free for the download from the University of Waterloo.

Stephenson's novel [16], though a work of fiction, is closely based upon real events surrounding cryptographic efforts in World War II and is an extremely good read for anyone with an interest in crypto.

The web site www.mersenne.org reflects the most up-to-date information on the search for ever larger Mersenne primes. There is a wealth of links on this site. Once you enter the "prime number web universe" you will encounter hundreds of sites that contain information on this subject.

Kahn's work on the Enigma [17] describes World War II events surrounding the Enigma, arguably the most powerful, efficient and field useful device ever produced for symmetric encryption prior to the advent of the digital computer. I have read that the Germans used prime number lists to assist in the generation of the "day code" books for the Enigma, but have been unable to locate confirmation of this.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

REFERENCES

[1] Stallings, W., *Cryptography and Network Security: Principles and Practice*, Second Edition, New Jersey: Prentice-Hall, 1999

[2] Schneier, B., *Applied Cryptography*, Second Edition, New York: John Wiley and Sons, 1996

[3] Schneier, B., *Secrets & Lies, Digital Security in a Networked World*, New York: John Wiley and Sons, 2000

[4] Singh, S., *The Code Book*, New York: Doubleday, 1999

[5] Miller, C., Lial, M. and Schneider, D., *Fundamentals of College Algebra*, 3rd Edition, Scott, Foresman and Company, 1990

[6] Diffie, W. and Hellman, M., "New Directions in Cryptography", *IEEE Transactions on Information Theory,* November, 1976

[7] Rivest, R., Shamir, A. and Adelman, L., "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", *Communications of the ACM*, February, 1978.

[8] Denning, D. and Denning, P., *Internet Besieged, Countering Cyberspace Scoflaws,* ACM Press (Addison-Wesley), New York, NY, 1998

[9] Denning, D., *Information Warfare and Security*, New York: Addison-Wesley/ACM Press, 1999

[10] Kahn, D., *The Codebreakers: The Story of Secret Writing*, New York: Macmillan, 1967; Scribner, 1996

[11] Menezes, A., van Oorschot, P. and Vanstone, S., *Handbook of Applied Cryptography*, CRC Press, 1996, pp 59-60, 63-65, 135-143; URL: http://www.cacr.math.uwaterloo.ca/hac/

[12] Manindra A., Kayal N. and Saxena, N., *PRIMES is in P*, unpublished. URL: http://www.cse.iitk.ac.in/news/primality.html]

[13] Peterson, Ivars, "Prime Pursuits", *Science News*, October 22, 2002.

[14] "Researchers Discover Largest Multi-Million-Digit Prime Using Entropia Distributed Computing Grid", URL: www.mersenne.org

[15] Various, "The Prime Pages", URL :www.utm.edu/research/primes (selected information about primes, particularly Mersenne primes).

[16] Stephenson, Neal, *Cryptonomicon*, New York: Perennial, 2000.

[17] Kahn, D., *Seizing the Enigma*, Boston: Houghton-Mifflin, 1991

[18] Lenstra, K. and Verheul, E., "Selecting Cryptographic Key Sizes", November, 1999, URL: http://www.pwcglobal.com/extweb/indissue.nsf/docid/AFF4D4A5F6BF6C13852568630074D015

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: May 23rd, 2013

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Malaysia @ MCMC 2013	Cyberjaya, MY	Jun 03, 2013 - Jun 08, 2013	Live Event
SANS Pen Test Berlin 2013	Berlin, DE	Jun 03, 2013 - Jun 08, 2013	Live Event
Industial Control Systems Security Training - Houston	Houston, TXUS	Jun 10, 2013 - Jun 15, 2013	Live Event
Security Impact of IPv6 Summit 2013	Washington, DCUS	Jun 14, 2013 - Jun 16, 2013	Live Event
SANSFIRE 2013	Washington, DCUS	Jun 14, 2013 - Jun 22, 2013	Live Event
SANS Canberra 2013	Canberra, AU	Jul 01, 2013 - Jul 13, 2013	Live Event
Digital Forensics & Incident Response Summit 2013	Austin, TXUS	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS London Summer 2013	London, GB	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS Rocky Mountain 2013	Denver, COUS	Jul 14, 2013 - Jul 20, 2013	Live Event
SANS Mumbai 2013	Mumbai, IN	Jul 22, 2013 - Jul 27, 2013	Live Event
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	Washington, DCUS	Jul 22, 2013 - Jul 26, 2013	Live Event
SANS San Francisco 2013	San Francisco, CAUS	Jul 29, 2013 - Aug 03, 2013	Live Event
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	Bangalore, IN	Aug 05, 2013 - Aug 10, 2013	Live Event
SANS Boston 2013	Boston, MAUS	Aug 05, 2013 - Aug 10, 2013	Live Event
Critical Security Controls Summit	Washington, DCUS	Aug 12, 2013 - Aug 18, 2013	Live Event
Industrial Control Systems Security Training - DC	Washington, DCUS	Aug 12, 2013 - Aug 16, 2013	Live Event
SANS Thailand 2013	Bangkok, TH	Aug 19, 2013 - Aug 31, 2013	Live Event
SANS Virginia Beach 2013	Virginia Beach, VAUS	Aug 19, 2013 - Aug 30, 2013	Live Event
Mobile Device Security Summit 2013	OnlineCAUS	May 30, 2013 - Jun 06, 2013	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced