**Week 6**

## 9.2 An Application to Cryptography

**Cryptography**– the study of the design and analysis of mathematical techniques that ensure secure communications in the presence of malicious adversaries – is the only known practical means for protecting information transmitted through public communications networks such as those using telephone lines, microwaves or satellites.

In the language of cryptography, where codes are called **ciphers**, the information to be concealed is called **plaintext**. After transformation to a secret form, a message is called **ciphertext**. The process of converting from plaintext to ciphertext is called **encrypting** or **enciphering**, while the reverse process of changing from ciphertext back to plaintext is called **decrypting** or **deciphering**.

One of the earliest cryptographic systems was used by the Roman emperor Julius Caesar around 50 B.C. He wrote to Marcus Cicero using a **simple substitution cipher** in which each letter of the alphabet is replaced by the letter which occurs three places down the alphabet, with the last three letters cycled back to the first three. If we write the ciphertext equivalent underneath the plaintext letter, the substitution alphabet for the **Caesar cipher** is:

| Plaintext: | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ciphertext: | D | E | F | G | H | I | J | K | L | M | N | O | P |

| Plaintext: | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ciphertext: | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |

Table 9.2: Substitution alphabet for the Caesar cipher

For example, the plaintext message "CAESAR WAS GREAT" is transformed into the ciphertext "FDHVDU ZDV JUHDW".

The Caesar cipher can be described very easily using congruences. Any plaintext is first expressed numerically by translating the characters of the text into digits by means of some correspondence like the following:

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 |

| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |

Table 9.3: Numerical expression of plaintext

If $P$ is the digital equivalent of a plaintext letter, and $C$ is the digital equivalent of the corresponding ciphertext letter, then

$$C \equiv P + 3 \pmod{26}.$$

For instance, the letters of the above message are converted to their equivalents

| C | A | E | S | A | R | W | A | S | G | R | E | A | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 03 | 01 | 05 | 19 | 01 | 18 | 23 | 01 | 19 | 07 | 18 | 05 | 01 | 20 |

Table 9.4: Caesar was great

Using the congruence $C \equiv P + 3 \pmod{26}$, this becomes the ciphertext 06 04 08 22 04 21 26 04 22 10 21 08 04 23. To recover the plaintext, the procedure is simply reversed by means of the congruence

$$P \equiv C - 3 \equiv C + 23 \pmod{26}.$$

The Caesar cipher is very simple and hence extremely insecure. Caesar himself soon abandoned this scheme, not only because of its insecurity, but also because he didn't trust Cicero, with whom he necessarily shared the secret of the cipher. *Any* substitution cipher is insecure, even if the substitution results from a random scrambling of the alphabet, because different letters occur with different frequencies, and by studying these frequencies and the possible positions of vowels, the secret is soon revealed, especially with the use of computers. In April 2006, fugitive Mafia boss Bernardo Provenzano was captured in Sicily partly because of cryptanalysis of his messages written in the Caesar cipher with a shift of 4.

In conventional cryptographic systems, such as the Caesar cipher, the sender and receiver jointly have a **secret key**. The sender uses the key to encrypt the plaintext to be sent, and the receiver uses the same key to decrypt the ciphertext obtained. **Public-key** cryptography differs from conventional cryptography in that it uses two keys, an **encryption key** and a **decryption key**. Although the two keys effect inverse operations and are therefore related, there is no easily computed method of deriving the decryption key from the encryption key. Thus the encryption key can be made public without compromising the decryption key; each user can encrypt messages, but only the intended recipient (whose decryption key is kept secret) can decipher them. A major advantage of a public-key cryptosystem is that it is unnecessary for each sender and receiver to exchange a key in advance of their decision to communicate with each other.

In 1977, R. Rivest, A. Shamir and L. Adleman proposed a public-key cryptosystem which uses only elementary ideas from number theory. Their enciphering system is called RSA, after the initials of the inventors. Its security depends on the assumption that in the current state of computer technology, the factorisation of composite numbers with large prime factors is prohibitively time-consuming, and MUCH slower than generating MUCH larger prime numbers.

Each user of the RSA system chooses a pair of distinct primes, $p$ and $q$, large enough that the factorisation of their product $n = pq$, called the **encryption modulus**, is beyond all current computational capabilities. For instance, the user – call him Bob – may pick $p$ and $q$ with approximately 200 digits each, so that $n$ has approximately 400 digits. Having selected $n$, Bob then chooses a random positive integer $k$, the **encryption exponent**, satisfying $(k, \phi(n)) = 1$. While there are many suitable choices for $k$, an obvious suggestion is to pick $k$ to be any prime larger that both $p$ and $q$. (This will ensure that $(k, \phi(n)) = 1$. Why?) The pair $(n, k)$ is placed in a public file, analogous to a telephone directory, as Bob's personal encryption key. This will allow anyone else – say Alice – in the communications network to encrypt and send a message to Bob. Notice that while $n$ is revealed, the listed public key does not mention the (private) factors $p$ and $q$.

The encryption process begins with Alice converting her message into an integer $M$ by means of a "digital alphabet" in which each letter, number or punctuation mark of the plaintext is replaced by a two-digit integer. One standard procedure is to use the assignment

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A = 01 | | K = 11 | | U = 21 | | 1 = 31 |
| B = 02 | | L = 12 | | V = 22 | | 2 = 32 |
| C = 03 | | M = 13 | | W = 23 | | 3 = 33 |
| D = 04 | | N = 14 | | X = 24 | | 4 = 34 |
| E = 05 | | O = 15 | | Y = 25 | | 5 = 35 |
| F = 06 | | P = 16 | | Z = 26 | | 6 = 36 |
| G = 07 | | Q = 17 | | , = 27 | | 7 = 37 |
| H = 08 | | R = 18 | | . = 28 | | 8 = 38 |
| I = 09 | | S = 19 | | ? = 29 | | 9 = 39 |
| J = 10 | | T = 20 | | 0 = 30 | | ! = 40 |

Table 9.5: Digital alphabet

with 00 indicating a space between words. In this scheme, the message

*The brown fox is quick.*

is transformed into the numerical string

$$M = 20080500021815231400061524000919001721090311 28.$$

It is assumed that the plaintext number $M$ is less than $n$, where $n$ is the encryption modulus, otherwise it would be impossible to distinguish $M$ from any larger integer congruent to $M$ (mod $n$). If the message is too long to be handled as a single number $M < n$, then $M$ can be broken up into blocks of digits $M_1$, $M_2$, ..., $M_t$ of the appropriate size. Each block would be sent separately.

Looking up Bob's encryption key $(n, k)$ in the public directory, Alice disguises the plaintext number $M$ as a ciphertext number $r$ by raising $M$ to the $k^{\text{th}}$ power and then reducing the result modulo $n$, that is,

$$M^k \equiv r \pmod{n},$$

where $r$ is a least residue of $n$. Thus $r$ is the ciphertext that is transmitted. A 200-character message can be encrypted in seconds on a computer. Recall that the public encryption exponent $k$ was originally selected so that $(k, \phi(n)) = 1$.

At the other end, Bob deciphers the transmitted information by first determining the integer $j$, the **secret recovery exponent**, for which

$$kj \equiv 1 \pmod{\phi(n)}.$$

Since $(k, \phi(n)) = 1$, this linear congruence has a unique solution modulo $\phi(n)$. In fact, the Euclidean algorithm will produce $j$ as a solution to $x$ in the equation

$$kx + \phi(n)y = 1.$$

The recovery exponent can be calculated only by someone who knows both $k$ and $\phi(n) = (p-1)(q-1)$, hence who knows the prime factors $p$ and $q$ of $n$. Thus, $j$ is secure from an illegitimate third party – Eve – whose knowledge is limited to the public-key $(n, k)$.

Bob can now retrieve $M$ from $r$ by simply calculating $r^j \pmod{n}$. Because $kj = 1 + \phi(n)t$ for some integer $t$, it follows from Euler's Theorem (Theorem 9.1) that

$$r^j \equiv (M^k)^j \equiv M^{1+\phi(n)t} \equiv M(M^{\phi(n)})^t \equiv M \cdot 1^t \equiv M \pmod{n},$$

whenever $(M, n) = 1$. In other words, raising the ciphertext number to the $j^{\text{th}}$ power and reducing it modulo $n$ recovers the original plaintext number $M$.

The assumption that $(M, n) = 1$ was made in order to use Euler's Theorem. In the unlikely (unlikely because $n$ is "almost" prime) event that $M$ and $n$ are not relatively prime, then either $p|M$ or $q|M$; assume $p|M$. Then $r^j \equiv (M^k)^j \equiv 0 \equiv M \pmod{p}$ and $r^j \equiv (M^k)^j \equiv M(M^{\phi(n)})^t \equiv M(M^{\phi(q)})^{\phi(p)t} \equiv M \pmod{q}$, so that $p|(M - r^j)$ and $q|(M - r_j)$. Since $(p, q) = 1$, $n|(M - r_j)$ and the desired congruence $r^j \equiv (M^k)^j \equiv M \pmod{n}$ follows.

**Cryptanalysis** is the process by which Eve, on receiving some ciphertext, determines the original message without prior knowledge of the (private) key $pq$. **Cryptology** is the study of both cryptography and cryptanalysis.

**Example:** (using small numbers to get an illustration that is easy to handle) Encrypt and then decrypt the message

NO EXIT

using the encryption modulus $n = 29 \cdot 53 = 1537$ and encryption exponent $k = 47$. (Thus the public-key is $(n, k) = (1537, 47)$). Note that $\phi(n) = \phi(29)\phi(53) = 28 \cdot 52 = 1456$, hence $(k, \phi(n)) = 1$ as required.

**Solution**

We begin by translating the message into its digital equivalent using the substitution mentioned earlier. This yields the plaintext number

$$M = 14150005240920.$$

We want each plaintext block to be an integer less than 1537. To be sure of this we split $M$ into blocks containing exactly one digit less than the encryption modulus, i.e. five 3-digit blocks, where we add a **0** at the end to fill the block.

$$M = 141 \;\; 500 \;\; 052 \;\; 409 \;\; 20\mathbf{0}.$$

The encryption is

$$\begin{aligned} 141^{47} &\equiv 658 \pmod{1537}, \\ 500^{47} &\equiv 1408 \pmod{1537}, \\ 052^{47} &\equiv 953 \pmod{1537}, \\ 409^{47} &\equiv 801 \pmod{1537}, \\ 200^{47} &\equiv 707 \pmod{1537}. \end{aligned}$$

Thus the ciphertext that is transmitted (in blocks of the same size – now it does not matter that the block size is the same as the number of digits of $n$, because the numbers are still least residues of $n$) is

$$0658 \;\; 1408 \;\; 0953 \;\; 0801 \;\; 0707.$$

The authorised recipient knows the secret recovery exponent $j$. It is the unique integer $j$ satisfying the congruence

$$kj \equiv 1 \pmod{\phi(n)},$$

and we know that $\phi(n) = 1456$. Thus we need the solution to

$$47j \equiv 1 \pmod{1456},$$

which is $j = 31$ (use the Euclidean algorithm). Hence the recipient decrypts the message as follows:

$$\begin{aligned} 658^{31} &\equiv 141 \pmod{1537}, \\ 1408^{31} &\equiv 500 \pmod{1537}, \\ 953^{31} &\equiv 52 \equiv \mathbf{0}52 \pmod{1537}, \\ 801^{31} &\equiv 409 \pmod{1537}, \\ 707^{31} &\equiv 200 \pmod{1537}. \end{aligned}$$

We must add a **0** in front of 52 (in front, not at the back, so as not to change 52) because we know each block has size exactly three. This gives the original message 141 500 052 409 200.

When substituting letters for digits, we simply discard the useless 0 (or string of 0's) at the end.

For the RSA cryptosystem to be secure it must not be computably feasible to recover the plaintext $M$ from the information assumed to be known to a third party, namely the listed public-key $(n, k)$. The direct method of attack would be to attempt to factor $n$, a huge integer, because once the factors are determined, the recovery exponent $j$ can be calculated from $\phi(n)$ and $k$. The confidence in the RSA system rests on the expected amount of computer time needed to factor the product of two large primes. Factoring is computationally MUCH more difficult than distinguishing between primes and composites. On today's fastest computers, a 200 digit number can routinely be tested for primality in less than 10 minutes, whereas the running time required to factor a composite number of the same size is prohibitive. It has been estimated that the quickest factoring algorithm known can use approximately $(1.2)10^{23}$ computer operations to resolve an integer with 200 digits into its prime factors. Assuming that each operation takes $10^{-6}$ seconds, the factorisation time would be about $(3.8)10^9$ years.

As an example, consider the Mersenne numbers. For $p$ prime, $\mathrm{M}(p)$ denotes the integer $2^p - 1$. If $\mathrm{M}(p)$ is prime, it is a Mersenne prime, otherwise it is a **Mersenne number**. There are 50 prime numbers $p$ for which it is known that $\mathrm{M}(p)$ is prime, the largest being $\mathrm{M}(77,232,917)$. For almost all prime numbers $p < 43,112,609$ it is known that $\mathrm{M}(p)$ is not prime. As of June 2018, M(1277) is the smallest composite Mersenne number with no known factors; it has no prime factors below $2^{67}$. This illustrates the enormous difference in the size of the largest known Mersenne prime and the size of the largest known composite Mersenne number whose factors are known.

When used in practice, RSA is generally combined with further enhancements to improve its security.

The above section on cryptography was adapted from D. M. Burton, *Elementary Number Theory* (fourth ed.), McGraw-Hill, New York, 1998. See Burton for other encryption algorithms.