SYNCHRONIZATION OF MULTI-CARRIER CDMA SIGNALS AND SECURITY ON INTERNET

BY

YOOH JI HENG

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF PHILOSOPHY

DIVISION OF INFORMATION ENGINEERING

THE CHINESE UNIVERSITY OF HONG KONG

JUNE 1996

Acknowledgement

First, I am grateful very much to my supervisor Dr. Wei Keh Wei who strengthen and broaden my knowledge in wireless communications and cryptography, and thanks for his guidance, patience, invaluable help and advice in my two years postgraduate studies.

Moreover, I would like to express my appreciation to all teaching staff in Department of Information Engineering of the Chinese University of Hong Kong especially to Dr. Liew Soung Chang and Dr. Wong Wing Shing who strengthen my knowledge in network through the courses I have tutored.

Many thanks to all technical staff and staff in general office in this department for their most kindness help. Besides, I want to acknowledge all my colleagues especially Choy Sai Chung, Ho Tsan Fai, Liu Chi Leung and Yeung Chi Kit.

I am grateful to Chan Lap Shing, Chan Wing Sze, Cheung Ying Kit, Chow Chiu Ling, Fan Chin Pang, Ip Hon Chung, Ip Wing Han, Jung Kwok Kwan, Lee Chi Sang, Ming Chui Ping, Pang Chun Lun, Yam Lai Hung and all my friends who always stand by me, support me and give me the most unforgettable days in my two years university life.

Special thanks to my parents for their patience and sincerest care, encouragement and support to me.

Abstract

In the first part of the thesis, we study two methods of synchronizing and detecting multi-carrier CDMA (MC-CDMA) signals. The first method uses the Fourier transform and a curve-fitting algorithm. This approach exploits the linearity between time shift and carrier phase shifts of MC-CDMA signals with equally spaced carriers. The second method is based on the matched filter. The peak values of the autocorrelation of the signal are detected to estimate the transmitted signal. The received waveform is first sampled and quantized. A fixed sampling clock can be used. No feedback mechanism, such as a PLL, is used to adjust the sample timing in either method. Moreover, we use Shapiro-Rudin sequence as the signature sequence to reduce the signal's peak-power-to-average-power ratio. We analyze the performances of these simple methods for MC-CDMA signals in an additive white Gaussian channel.

In the second part of the thesis, we study some security aspects in Internet and Internet commerce systems. First we start with a brief introduction of mathematical theories behind the security systems. Then some well known private and public key cryptographic algorithms are presented. Some security softwares which employ these algorithms are also discussed. Using these security softwares and algorithms, we can establish infrastructures of some Internet commerce systems. The structures include some popular Internet browsers, different electronic payment methods like credit card, electronic cash and cheque, and different commercial payment systems including CyberCash, DigiCash, Mondex and so on.

Contents

List of Figures

Part I

Synchronization of Multi-carrier CDMA Signals

Chapter 1

Introduction

Spread Spectrum Code Division Multiple Access (SS-CDMA) has shown to be a good candidate for modern broadband mobile communications. It has good immunity to noise and fading, and reasonable privacy and security. Common forms of SS-CDMA include (Direct Sequence) **DS/SS-CDMA**, (Frequency Hopping) **FH/SS-CDMA** and (Time Hopping) **TH/SS-CDMA**. TH/SS-CDMA is mainly for military use whereas DS/SS-CDMA and FH/SS-CDMA are commonly used for commercial purposes.

An alternative broadband CDMA technique is the **Multi-carrier CDMA** (MC-CDMA) system. It uses the idea of **Orthogonal Frequency Division Modulation** (OFDM) with CDMA [2, 3, 4]. Multi-carrier systems have potential advantages over single-carrier system [6, 7]. An interesting duality exists between DS-CDMA and MC-CDMA [2, 3]. For SS-CDMA, the spreading sequence is applied to the signal in time. On the contrary, MC-CDMA system multiplies the signature sequence to different carriers in frequency.

Like other multi-carrier systems, MC-CDMA also has the problem of large peak-power-to-average-power ratio. To solve it, a binary sequence called the **Shapiro-Rudin sequence** can be used as the signature sequence [14].

In digital communications, synchronization of carrier phase and frequency is very important. Conventional methods use feedback mechanism, such as the **Phase Locked Loop** (PLL), to perform acquisition and tracking of the carrier phase and frequency. In a multi-carrier system, using multiple PLLs for the multiple carriers is expensive. Therefore, we study two other approachs which do not require the PLL [16].

The first approach exploits the linearity between the time shift and carrier phase shifts of MC-CDMA signals with equally spaced carriers. The **FFT** is performed to extract the carrier coefficients. Then the despreading operation is performed on these coefficients, followed by a curve-fitting algorithm to estimate the signal's time shift. The resulting vector of coefficients is used to detect the signal. The second approach is based on matched filter. It evaluates the autocorrelation of the MC-CDMA signal. Peak values at the appropriate moments are used to estimate the transmitted signal. In either case, no feedback mechanism is used to adjust the sample timing. A fixed sampling clock can be used, thus simplifying the circuitry.

1.1 Spread Spectrum CDMA

In some common multiple access methods like Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA), there exist problems of multipath fading and interference, many researches have already shown that Spread Spectrum CDMA (SS-CDMA) is a good alternative to them [8].

The main features of SS-CDMA includes [8, 9]:

1. Good immunity of jamming and interference.
2. As spread spectrum means that the signal will occupy a wide bandwidth and introduce a frequency diversity, hence SS-CDMA can achieve a good performance in multipath fading channel due to multiple scatterers with different delays (delay spread).
3. Each user has a signature code (PN sequence), it enables privacy and security.
4. **Universal frequency reuse** - in TDMA and FDMA, the assigned frequencies in neighboring cells is different to reduce co-channel interference. In SS-CDMA, the allocated spectrum can be used simultaneously by users in every cells which also means that the frequency reuse factor is 1 and frequency planning is much simple.
5. **Soft handoff** - applying the RAKE receiver and the frequency reuse property, a mobile station is able to receive the same signal from two adjacent base stations. A new connection can make before the old connection breaks

and it is called soft handoff. In this case, the probability of a call is dropped is reduced and better reception at cell's boundary can be achieved.

Based on the features above, the cell capacity of SS-CDMA is higher. However, there are some difficulties that SS-CDMA system faces: (i) accurate power control to solve the "near-far" problem in reverse link and (ii) the timing of the pilot of DS/SS-CDMA signal for synchronization.

1.1.1 Direct Sequence/SS-CDMA

In (Direct Sequence) DS/SS-CDMA, a bit interval is divided into N chips with each chip of duration $T_c = T_b/N$. Suppose the information sequence $a_k \in \{-1, 1\}$, then each chip in an information bit will be multiplied by a pseudo-random (PN) sequence $c_i \in \{-1, 1\}$ of length N. Since $T_c = T_b/N$, so the signal spectrum is spread by a factor of N. The use of RAKE receiver can improve the performance by constructively combining multipath components with time delays greater than the signal correlation time [8].

1.1.2 Frequency Hopping/SS-CDMA

In (Frequency Hopping) FH/SS-CDMA, like that in DS/SS-CDMA, the bit interval is also divided into N chips of duration $T_c = T_b/N$. In each bit interval, the signal is transmitted at different frequencies at different chips (the signal is hopped over different frequencies). The frequency synthesizer in the transmitter

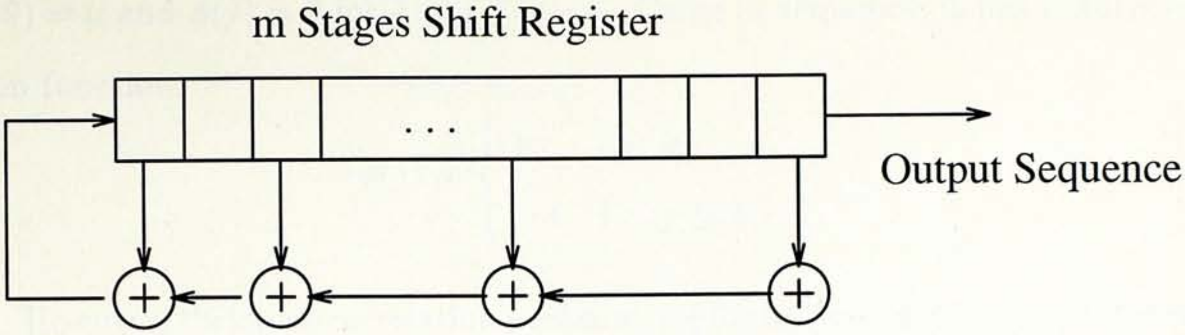


Figure 1.1: The Block Diagram of a m-sequence Generator

will use the PN sequence to determine the frequency used for each chip. FH/SS-CDMA has an advantage over DS/SS-CDMA that the frequency spectrum do not need to be contiguous. However, the frequency hopping pattern should be carefully selected to maintain orthogonality.

1.1.3 Pseudo-noise Sequence

One of the most commonly known binary (pseudo-noise/pseudo-random) PN sequences is the maximum-length shift register sequence (**m-sequence**) [10]. An m-sequence is a periodic sequence generated by an m-stage shift register as shown in Fig 1.1. The length and the period of an m-sequence is $n = 2^m - 1$ with 2^{m-1} ones and $2^{m-1} - 1$ zeros. The m-sequence $b_i \in \{0, 1\}$ obtained is mapped into a binary sequence $c_i \in \{-1, 1\}$ before multiplying to the input signal.

The autocorrelation of a PN sequence with period n is,

$$\phi(j) = \sum_{i=1}^{n}(2b_i - 1)(2b_{i+j} - 1) \qquad\qquad 0 \le j \le n - 1.$$

For ideal pseudo-random sequence, the autocorrelation function should be

$\phi(0) = n$ and $\phi(j) = 0$ for $1 \leq j \leq n-1$. Using m-sequence, it has autocorrelation function,

$$\phi(j) = \begin{cases} n & j = 0 \\ -1 & 1 \leq j \leq n-1. \end{cases}$$

However, the cross-correlation performance for m-sequence is not good enough for some CDMA applications. Some other PN sequences are proposed with better cross-correlation characteristic. For example, the Gold sequences and Kasami sequences derived from certain m-sequences have a three-valued cross-correlation function [10].

1.2 Synchronization for CDMA signal

1.2.1 Acquisition of PN Sequence

For DS/SS-CDMA signal, the main synchronization problem is the synchronization of the spreading and despreading PN sequences. This process mainly involves two stages [13], the *PN acquisition* stage and the *PN tracking* stage. Initailly, in the PN acquisiton stage, phase difference between the two sequences will be adjusted to within a certain small time offset. Then the two sequences will maintain synchronized in the PN tracking stage by a closed loop like Delay-Locked Loop (DLL) or Tau-Dither Loop (TDL).

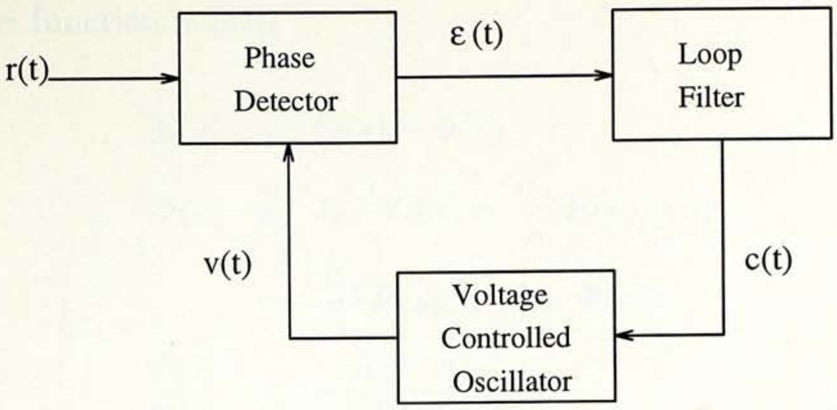


Figure 1.2: The Block Diagram of a Phase Lock Loop

1.2.2 Phase Locked Loop

Phase Locked Loop (PLL) is a widely-used technique for acquisition and tracking of the carrier phase and frequency [11]. The basic structure of a PLL is consist of 3 main components, *phase detector*, *the loop filter* and *voltage-controlled oscillator* (VCO) Fig.1.2. For a received signal $r(t)$, the phase detector's output $\varepsilon(t)$ is a function of the phase difference between $r(t)$ and the VCO output $v(t)$. Then $\varepsilon(t)$ will enter the loop filter to obtained the control signal $c(t)$ for the VCO.

Suppose the received signal is $r(t) = \cos(2\pi f_o t + \theta(t))$ and the output of the VCO is $v(t) = \cos(2\pi f_o t + \phi(t))$ [10, 12]. If the phase error is small such that the phase detector can operate in linear range, $\varepsilon(t) = \theta(t) - \phi(t)$. The phase of the VCO output $\phi(t)$ is depended on the control signal $c(t)$,

$$\phi(t) = K_v \int_{-\infty}^{t} c(\tau)d\tau,$$

where K_v is the VCO gain constant. The loop filter is a low-pass filter which filter out the high frequency components in the phase error $\varepsilon(t)$ and it has a transfer function $F(s)$. Applying Laplace transform to the equations above, the phase transfer function is [11],

$$\begin{aligned}
E(s) &= \Theta(s) - \Phi(s) \\
\Phi(s) &= K_v \frac{1}{s} C(s) = \frac{K_v}{s} E(s) F(s) \\
&= \frac{K_v}{s} F(s)(\Theta(s) - \Phi(s)) \\
\frac{\Phi(s)}{\Theta(s)} &= \frac{K_v F(s)}{s + K_v F(s)}.
\end{aligned}$$

If a second order loop filter is used, the transfer function $F(s)$ has the form,

$$F(s) = \frac{1 + K_2 s}{1 + K_1 s} \qquad (K_1 \gg K_2).$$

The phase transfer function is,

$$\begin{aligned}
\frac{\Phi(s)}{\Theta(s)} &= \frac{K_v \left(\dfrac{1 + K_2 s}{1 + K_1 s}\right)}{s + K_v \left(\dfrac{1 + K_2 s}{1 + K_1 s}\right)} \\
&= \frac{K_v(1 + K_2 s)}{K_v + (1 + K_v K_2) s + K_1 s^2}.
\end{aligned}$$

Chapter 2

Multi-carrier CDMA

In DS/SS-CDMA, the bit duration is divided into many chips and the spreading/signature sequence is applied to each chip in time domain. The same idea can be applied in the frequency domain to a signal using OFDM (**Orthogonal Frequency Division Modulation**) or MCM (**Multi-carrier Modulation**). The entire frequency band is divided into many subbands and each subband has a carrier which is orthogonal to each other [1, 3]. In contrast to DS/SS-CDMA, the spreading sequence is multiplied to each carrier frequency.

To construct a MC-CDMA signal with n carriers, the information sequence will enter an n-point inverse Discrete Fourier Transform (IDFT) block with the spreading/signature sequence multiplied to each carrier. Then the resulting sequence is LPF and the output continous waveform is transmited. At the receiver, the received signal will enter an n-point FFT block with the inverse of the spreading (despreading) sequence multiplied to each carrier as in the transmitter [5]. The use of IFFT and FFT shows the duality relation between DS/SS-CDMA and MC-CDMA.

MC-CDMA can be considered as an alternative technique of current SS-CDMA system. Comparing with FH/SS-CDMA, MC-CDMA does not require powerful synchronization and the receiver is less complex since frequency hopping is not need. Also, MC-CDMA does not need channel estimation and equalization as in DS/SS-CDMA [1]. Moreover, it is known that *doppler spread* is a dual of *delay spread*, and the RAKE receiver is used for DS/SS-CDMA in multipath channel with delay spread. This duality shows that the RAKE receiver can be used for MC-CDMA in multipath channel due to scattering on multiple moving objects (doppler spread) [3]. However, problems of MC-CDMA include the sensitivity of non-linearity in power amplifier and carrier synchronization [5].

2.1 System Model

An MC-CDMA transmitter is shown in Fig. 2.1. The information bit $a_k \in \{-1,1\}$ $(-\infty < k < \infty)$ is multiplied by the spreading sequence $\{c_n\} \in \{-1,1\}$ before entering a serial-to-parallel converter. The parallel output is multiplied to the carriers and then summed to constitute the following MC-CDMA signal (before up-conversion)

$$s(t) = \sum_{k=-\infty}^{\infty} a_k p(t - kT_b),$$

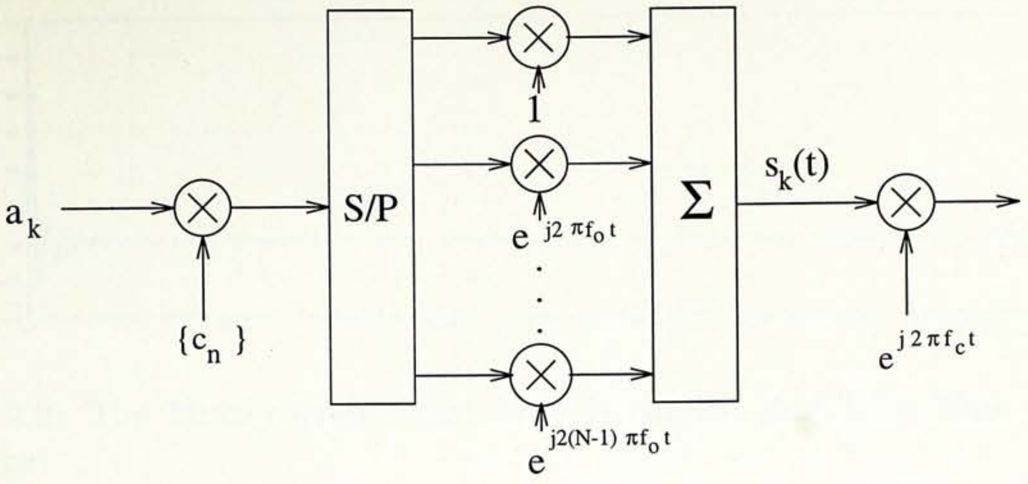


Figure 2.1: The Block Diagram of the Transmitter of an MC-CDMA System

where

$$p(t) = rect(\frac{t}{T_b}) \sum_{n=0}^{N-1} c_n e^{j2\pi n f_o t},$$

$rect(\cdot)$ is the rectangular function, N is the number of carriers, $f_o = 1/T_b$ is the frequency spacing between adjacent carriers, and T_b is the bit duration of the signal.

2.2 Crest Factor

A potential problem of MC-CDMA signals is the large ratio of peak power to average power. This ratio is often called the *crest factor* of the signal waveform $s(t)$ [14].

$$CF\{s(t)\} = \frac{\| s(t) \|_\infty}{\| s(t) \|_2},$$

where $\| s(t) \|_\infty$ means the maximum magnitude of the signal $s(t)$ and $\| s(t) \|_2$ is the root-mean-square value of $s(t)$,

$$\| s(t) \|_\infty = \sup_t |s(t)|,$$

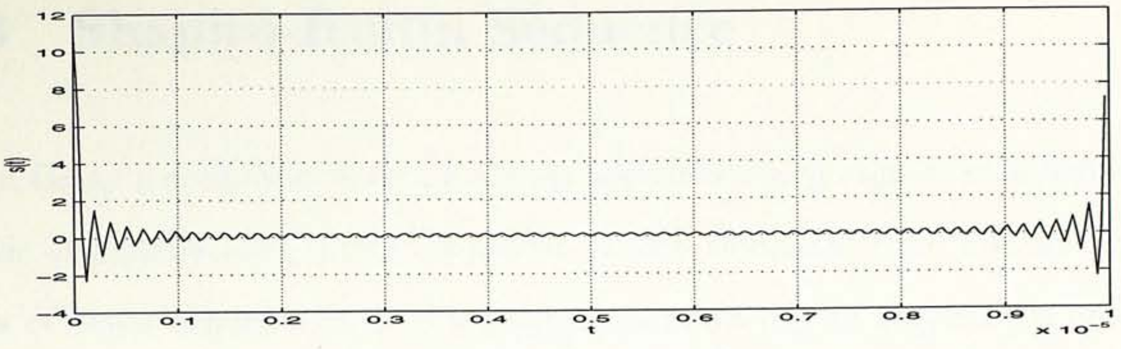


Figure 2.2: The Multi-carrier signal with 64 carriers and $T_b = 10\mu s$ (All one Sequence)

and

$$\| s(t) \|_2 = \left\{ \frac{1}{T_b} \int_0^{T_b} s^2(t)\, dt \right\}^{\frac{1}{2}} .$$

The maximum magnitude of $s(t)$ is found depending on the choice of the signature sequence $\{c_n\}$. Random or pseudo-random selections of the spreading sequence $\{c_n\}$ often results in large crest factors which may grow without bound as N (the number of carriers) increases. A normalised multicarrier signal using all one sequence and random sequence is shown in Fig. 2.2 and Fig. 2.3 respectively, the crest factor is the maximum amplitude of the signal. However, if $\{c_n\}$ is selected from certain classes of sequences, such as the Shapiro-Rudin sequences or the Golay complementary sequences, then the crest factor (measured in the baseband) is no more than 2 (or 6 dB), for any value of N [14]. A normalised multicarrier signal using Shapiro-Rudin sequence is shown in Fig. 2.4.

2.3 Shapiro-Rudin Sequence

From Golay's definition, a set of binary *complementary sequences* is defined as a pair of equally long, finite sequences of two elements [15]. The number of pairs of equal elements with any given separation in one sequence is equal to the number of pairs of unequal elements with the same separation in the other sequence.

This basic property can be represented by the characteristic that the sum of autocorrelations of the two sequences is zero except at zero shift. For two complementary sequences A and B with length n and elements {-1,1}, they have autocorrelation functions given by,

$$\phi_a(j) = \sum_{i=1}^{n} a_i a_{i+j} \quad \text{and} \quad \phi_b(j) = \sum_{i=1}^{n} b_i b_{i+j}$$
$$0 \leq j \leq n-1$$

where,

$$\phi_a(j) + \phi_b(j) = \begin{cases} 2n & j = 0 \\ 0 & 1 \leq j \leq n-1. \end{cases}$$

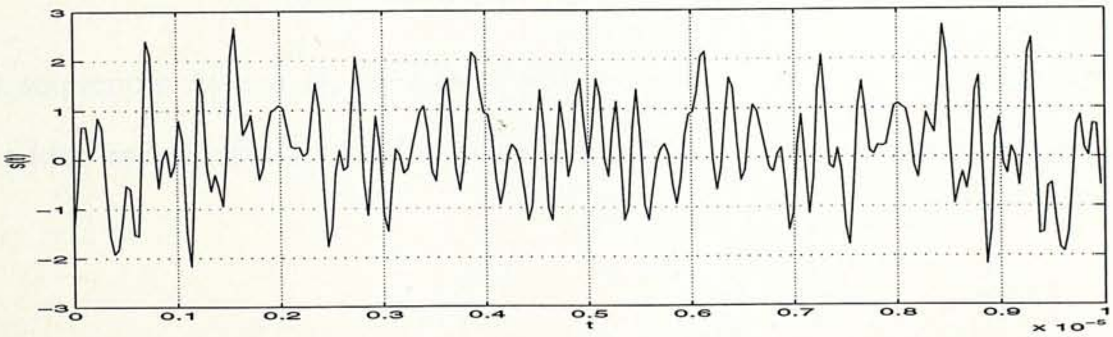


Figure 2.3: The Multi-carrier signal with 64 carriers and $T_b = 10\mu s$ (Random Sequence)

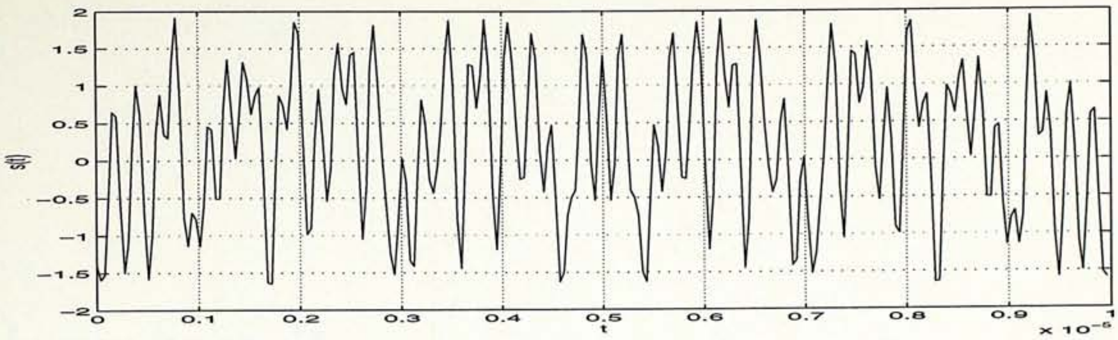


Figure 2.4: The Multi-carrier signal with 64 carriers and $T_b = 10\mu s$ (Shaprio-Rudin Sequence)

For binary complementary sequence, the number of elements must be even and a sum of at most two squares. If we construct two binary complementary sequences in which N is a power of 2, the resulting sequences are known as the **Shapiro-Rudin sequences**. Suppose we start from two Shapiro-Rudin sequences of length 4,

$$A = \{1 - 1 - 1 - 1\} \quad \text{and} \quad B = \{-11 - 1 - 1\},$$

we can construct others Shapiro-Rudin sequences with length 8 by appending the sequences A and B, and the sequences A and B' (complement of B),

$$A_1 = AB = \{1 - 1 - 1 - 1 - 11 - 1 - 1\}$$

$$B_1 = AB' = \{1 - 1 - 1 - 11 - 111\}.$$

Both sequences A_1 and B_1 are also complementary sequences. We can repeat this procedure recursively to further obtain any Shapiro-Rudin sequence of length 2^N.

Chapter 3

Synchronization and Detection by Line-Fitting

3.1 Unmodulated Signals

We consider the synchronization acquisition stage. Assume an unmodulated synchronization preamble, $\{a_k\} = \{+, +, ..., +\}$, is sent. The transmitted waveform is

$$s(t) = \sum_{n=0}^{N-1} c_n e^{j2\pi n f_o t}.$$

Assuming the signal suffers a time shift of τ, the received waveform, after down-conversion, is

$$r(t) = \sum_{n=0}^{N-1} c_n e^{j[\theta + 2\pi n f_o (t+\tau)]} + n(t).$$

The term $n(t)$ encompasses noise and multiple-access interferences. We assume the transmit baseband signal $s(t)$ is up-shifted to a higher carrier band f_c for transmission and then downshifted back to baseband at the front end of the receiver. We further assume that the receiver frequency down shifting requires no carrier phase synchronization, then carry phase shift of θ in the received baseband signal $r(t)$. Let the sample value of $r(t)$ be $(r_0, r_1, \cdots, r_{N-1})$ with

$$r_k = \sum_{n=0}^{N-1} c_n e^{j[\theta + 2\pi n f_o (\frac{kT_b}{N} + \tau)]} + n_k.$$

We model the noise term $n_0, n_1, \cdots, n_{N-1}$ as i.i.d. Gaussian noise. This simplifies our derivations. Further detailed studies may be needed to justify this assumption. We do not discuss them here. After Fourier transform and multiplying the resulting Fourier coefficients by $\{1/c_n\}$, we obtain the coefficient vector (R_0, R_1, ..., R_{N-1}) where

$$\begin{aligned}
R_n &= \frac{1}{c_n N} \sum_{k=0}^{N-1} r_k e^{-j2\pi \frac{kn}{N}} \\
&= \frac{1}{c_n N} \sum_{k=0}^{N-1} \left(\sum_{m=0}^{N-1} c_m e^{j[\theta + 2\pi m(\frac{k}{N} + f_o \tau)]} + n_k \right) e^{-j2\pi \frac{kn}{N}} \\
&= \frac{1}{c_n N} \sum_{m=0}^{N-1} c_m e^{j[\theta + 2\pi m f_o \tau]} \sum_{k=0}^{N-1} e^{j2\pi k(\frac{m-n}{N})} + \frac{1}{c_n N} \sum_{k=0}^{N-1} n_k e^{-j2\pi \frac{kn}{N}} \\
&= \frac{1}{c_n N} c_n e^{j[\theta + 2\pi n f_o \tau]} \sum_{k=0}^{N-1} e^0 + \frac{1}{c_n N} \sum_{\substack{m=0 \\ m \neq n}}^{N-1} c_m e^{j[\theta + 2\pi m f_o \tau]} \sum_{k=0}^{N-1} e^{j2\pi k(\frac{m-n}{N})} + \epsilon_n \\
&= e^{j[\theta + 2\pi n f_o \tau]} + \frac{1}{c_n N} \sum_{\substack{m=0 \\ m \neq n}}^{N-1} c_m e^{j[\theta + 2\pi m f_o \tau]} \left[\frac{1 - e^{j2\pi(m-n)}}{1 - e^{j2\pi(\frac{m-n}{N})}} \right] + \epsilon_n.
\end{aligned}$$

Therefore,

$$R_n = e^{j[\theta + 2\pi n f_o \tau]} + \epsilon_n \quad \text{where} \quad \epsilon_n = \frac{1}{c_n N} \sum_{k=0}^{N-1} n_k e^{-j2\pi \frac{kn}{N}}.$$

See Figure 3.1. Let $\phi_n = \angle R_n$. Typical values of $\{\phi_n\}$ are plotted in Figure 3.2 (without θ and ϵ_n). The graph is a line with slope $2\pi\tau$, except discontinuities involving 2π phase jumps.

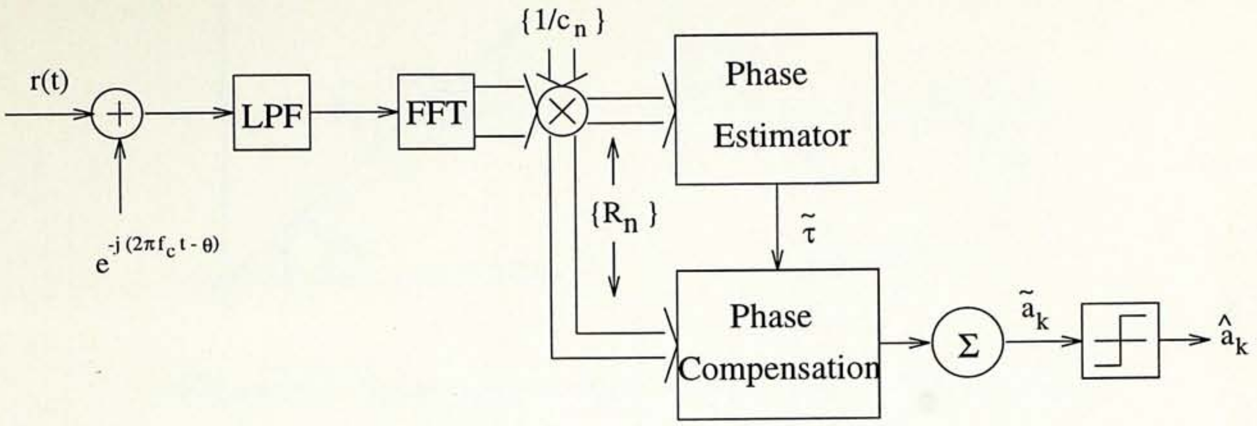


Figure 3.1: The Block Diagram of the Phase Compensation Model

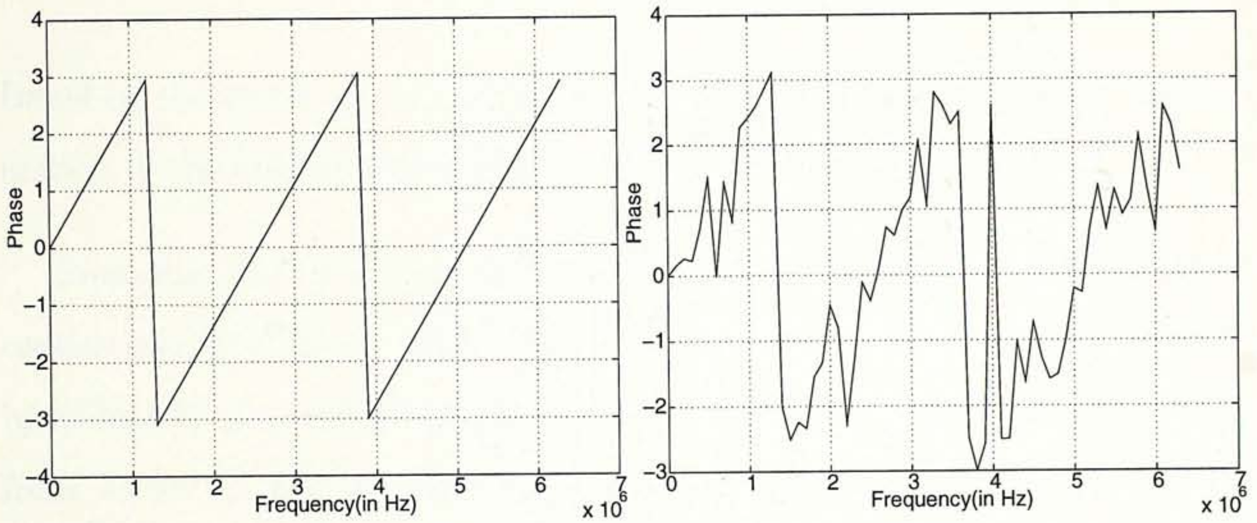


Figure 3.2: The Phase Spectrum without (left) and with AWGN (right) for $\tau = 0.391\mu s$

$$\phi_n = \begin{cases} 2\pi n f_o \tau \pmod{2\pi} & \text{if } 2\pi n f_o \tau \pmod{2\pi} < \pi \\ (2\pi n f_o \tau \pmod{2\pi}) - 2\pi & \text{otherwise.} \end{cases}$$

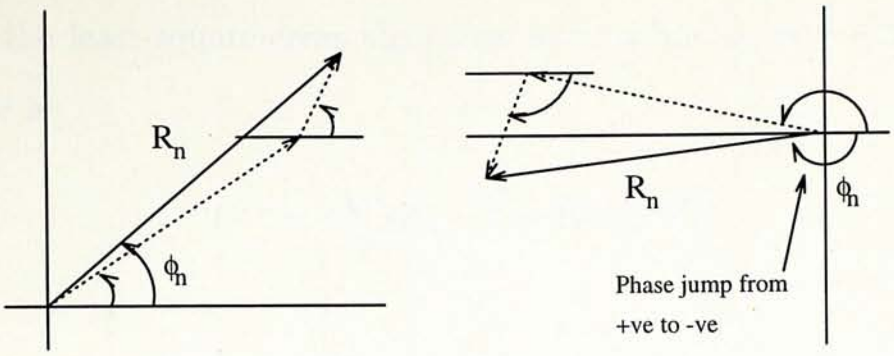


Figure 3.3: Example of sudden Phase Jump due to Noise

3.2 Estimating the Time Shift by Line-Fitting

Based on the vector $(\phi_0, \phi_1, ..., \phi_{N-1})$, the phase estimator obtains an approximation to the unknown time shift τ and the carrier phase shift θ.

From Fig. 3.2, we can see that the phase spectrum has slope $2\pi\tau$ except discontinuities involving 2π phase jumps. We compensate for these discontinuities by adding 2π after every jump from π to $-\pi$ (or subtracting 2π after every jump from $-\pi$ to π). Let the resulting phases be denoted $\{\tilde{\phi}_n\}$. Then a line-fitting algorithm is used. The slope of the fitted line is $\tilde{\tau}$, an estimate of the time shift τ.

When noise is added, the jump may not be exactly 2π. Therefore, in the simulation, we add 2π after every jump $> -\pi$ (or subtract 2π after every jump $> \pi$). Moreover, consider the phase vector $\{\phi_n\}$ of the phase spectrum without noise. For those $\phi_i \approx \pi$ (or $-\pi$), small noise added may alter them to negative (positive) values, i.e. large phase jumps occur. In this case, the 2π compensation should not apply to these values. The difficulty to overcome this problem affects the performance of the line-fitting especially for large τ and small SNR.

We use the least-square-error algorithm to fit a line $\phi_n = \tilde{\theta} + 2\pi n f_o \tilde{\tau}$. The square error is

$$E^2 = \sum_n (\phi_n - \tilde{\theta} - 2\pi n f_o \tilde{\tau})^2.$$

Solving for,

$$\begin{aligned} \partial E^2/\partial\tilde{\theta} &= -2\sum_n (\phi_n - \tilde{\theta} - 2\pi n f_o \tilde{\tau}) = 0 \\ \partial E^2/\partial\tilde{\tau} &= -4\pi \sum_n n f_o (\phi_n - \tilde{\theta} - 2\pi n f_o \tilde{\tau}) = 0. \end{aligned}$$

We obtain,

$$\begin{cases} 2\pi\tilde{\tau}\sum_{n=0}^{N-1} n f_o + \tilde{\theta} N & = \sum_{n=0}^{N-1} \phi_n \\ 2\pi\tilde{\tau}\sum_{n=0}^{N-1} n^2 f_o^2 + \tilde{\theta}\sum_{n=0}^{N-1} n f_o & = \sum_{n=0}^{N-1} n f_o \phi_n. \end{cases}$$

We can simplify the summation in the equations above.

$$\begin{aligned} \sum_{n=0}^{N-1} n f_o &= \frac{N(N-1)}{2} f_o \\ \sum_{n=0}^{N-1} n^2 f_o^2 &= \frac{N(N-1)(2N-1)}{6} f_o^2. \end{aligned}$$

Solving the system of equations above, we have

$$\begin{aligned} \tilde{\tau} &= \frac{N\sum_{n=0}^{N-1} n f_o \phi_n - \sum_{n=0}^{N-1} n f_o \sum_{n=0}^{N-1} \phi_n}{2\pi[N\sum_{n=0}^{N-1} n^2 f_o^2 - (\sum_{n=0}^{N-1} n f_o)^2]} \\ &= \frac{N f_o \sum_{n=0}^{N-1} n\phi_n - N(N-1)f_o/2\sum_{n=0}^{N-1}\phi_n}{N^2(N-1)(2N-1)\pi f_o^2/3 - N^2(N-1)^2\pi f_o^2/2} \\ &= \frac{6}{N(N-1)f_o[(4N-2)-3(N-1)]}\left[\sum_{n=0}^{N-1} n\phi_n - \frac{(N-1)}{2}\sum_{n=0}^{N-1}\phi_n\right] \\ &= \frac{3}{N(N^2-1)\pi f_o}\left[2\sum_{n=0}^{N-1} n\phi_n - (N-1)\sum_{n=0}^{N-1}\phi_n\right], \\ \tilde{\theta} &= \frac{\sum_{n=0}^{N-1}\phi_n \sum_{n=0}^{N-1} n^2 f_o^2 - \sum_{n=0}^{N-1} n f_o \sum_{n=0}^{N-1} n f_o \phi_n}{N\sum_{n=0}^{N-1} n^2 f_o^2 - (\sum_{n=0}^{N-1} n f_o)^2} \end{aligned}$$

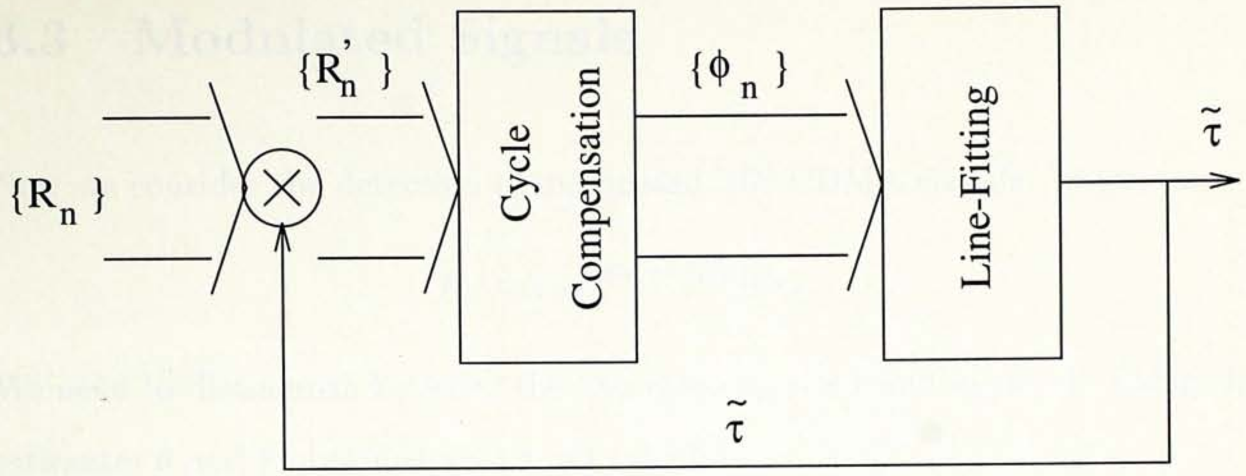


Figure 3.4: The Block Diagram of the Phase Estimator

$$= \frac{12}{N^2(N^2-1)f_o^2}\left[\frac{N(N-1)(2N-1)f_o^2}{6}\sum_{n=0}^{N-1}\phi_n - \frac{N(N-1)f_o^2}{2}\sum_{n=0}^{N-1}n\phi_n\right]$$

$$= \frac{2}{N(N+1)}\left[(2N-1)\sum_{n=0}^{N-1}\phi_n - 3\sum_{n=0}^{N-1}n\phi_n\right].$$

These two values achieve the least mean square error of phases $\{\phi_n\}$. They do not constitute a least mean square error estimate of the Fourier coefficients $\{R_n\}$.

In the simulation, we iterate the line-fitting several times. After each iteration, we deduct the estimated $\tilde{\tau}$ time shift from the next $\{R_n\}$ before the line fitting (Fig. 3.4). The new Fourier coefficients $\{R'_n\}$ are

$$R'_n = e^{j[\theta+2\pi n f_o(\tau-\tilde{\tau})]} + \epsilon_n e^{-j2\pi n f_o\tilde{\tau}}.$$

In this way, the slope of the phase spectrum will reduce and hence less 2π phase jumps will appear. For noisy channels, we find multiple iterations improve the performance especially for large τ. However, the estimated $\tilde{\theta}$ is not used to adjust the carrier phase syncrhonization. This simplifies the carrier phase synchronization procedure and does not adversely affect the algorithm performance.

3.3 Modulated Signals

Now we consider the detection of modulated MC-CDMA signals. In this case,

$$R_n = a_k e^{j[\theta + 2\pi n f_o \tau]} + \epsilon_n.$$

We need to distinguish between the two cases $a_k = +1$ and $a_k = -1$. Using the estimates $\tilde{\theta}$ and $\tilde{\tau}$ obtained above, we calculate

$$\begin{aligned} \tilde{a}_k &= \frac{1}{N} \sum_{n=0}^{N-1} R_n e^{-j[\tilde{\theta} + 2\pi n f_o \tilde{\tau}]} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} \left\{ a_k e^{j[2\pi n f_o(\tau - \tilde{\tau}) + \theta - \tilde{\theta}]} + \epsilon_n e^{-j[2\pi n f_o \tilde{\tau} + \tilde{\theta}]} \right\}. \end{aligned}$$

Then the decision symbol $\hat{a}_k$ is taken to be the sign of the estimate symbol $\tilde{a}_k$. See Figure 3.1.

Assume each ϵ_n is an i.i.d. Gaussian random variable with zero mean and variance

$$\overline{|\epsilon_n|^2} = \overline{|n(t)|^2} = \sigma^2,$$

then the signal-to-noise ratio is

$$\begin{aligned} SNR &= (|\sum_{n=0}^{N-1} e^{j2\pi n f_o(\tau - \tilde{\tau})}|^2)/(N\sigma^2) = \frac{1}{N\sigma^2} \left| \frac{e^{j2\pi N f_o(\tau-\tilde{\tau})} - 1}{e^{j2\pi f_o(\tau-\tilde{\tau})} - 1} \right|^2 \\ &= \frac{1}{N\sigma^2} \left| \frac{(e^{j\pi N f_o(\tau-\tilde{\tau})} - e^{-j\pi N f_o(\tau-\tilde{\tau})})/j2\pi}{(e^{j\pi f_o(\tau-\tilde{\tau})} - e^{-j\pi f_o(\tau-\tilde{\tau})})/j2\pi} \times \frac{j2\pi e^{j\pi N f_o(\tau-\tilde{\tau})}}{j2\pi e^{j\pi f_o(\tau-\tilde{\tau})}} \right|^2 \\ &= \frac{1}{N\sigma^2} \left| \frac{\sin(\pi N f_o(\tau - \tilde{\tau}))}{\sin(\pi f_o(\tau - \tilde{\tau}))} e^{j\pi(N-1) f_o(\tau-\tilde{\tau})} \right|^2 \\ &= \frac{\sin^2(\pi N f_o(\tau - \tilde{\tau}))}{N\sigma^2 \sin^2(\pi f_o(\tau - \tilde{\tau}))}. \end{aligned}$$

Note that if the time shift estimate is accurate, i.e. $\tilde{\tau} = \tau$, then the signal-to-noise ratio is N/σ^2, realizing the spread spectrum processing gain.

Chapter 4

Matched Filter

The transversal (i.e. discrete-time) matched filter is often used in the detection of DS-CDMA signals. It can be used in MC-CDMA also. It is based on the characteristics of the autocorrelation of the signalling pulse $p(t)$. The discrete-time autocorrelation function with ηN complex-valued samples per T_b (where η is a positive integer) is

$$\begin{aligned} a(\tau) &= \sum_{i=1}^{\eta N} p^*(\frac{iT_b}{\eta N})p(\frac{iT_b}{\eta N}+\tau) \\ &= \sum_i \sum_n \sum_{n'} c_n^* c_{n'} e^{j2\pi(n'-n)i/(\eta N)-j2\pi n f_o \tau} \\ &= \sum_n \sum_{n'} c_n^* c_{n'} e^{-j2\pi n f_o \tau} \sum_{i=1}^{\eta N} e^{j2\pi(n'-n)i/(\eta N)} \end{aligned}$$

if $n = n'$,

$$\sum_{i=1}^{\eta N} e^{j2\pi(n-n')i/(\eta N)} = \sum_{i=1}^{\eta N} e^0 = \eta N$$

if $n \neq n'$ (Since $\eta > 1$, $n - n' < N - 1 < \eta N/2$),

$$\sum_{i=1}^{\eta N} e^{j2\pi(n-n')i/(\eta N)} = e^{j2\pi(n-n')/(\eta N)} \frac{e^{j2\pi(n-n')} - 1}{e^{j2\pi(n-n')/(\eta N)} - 1} = 0.$$

So,

$$a(\tau) = \eta N \sum_{n=0}^{N-1} |c_n|^2 e^{-j2\pi n f_o \tau}.$$

If $|c_n| = 1$,

$$\begin{aligned} a(\tau) &= \eta N \sum_{n=0}^{N-1} e^{-j2\pi n f_o \tau} = \eta N \left[\frac{1 - e^{-j2\pi N f_o \tau}}{1 - e^{-j2\pi f_o \tau}} \right] \\ &= \eta N \left[\frac{(e^{j\pi N f_o \tau} - e^{-j\pi N f_o \tau})/j2\pi}{(e^{j\pi f_o \tau} - e^{-j\pi f_o \tau})/j2\pi} \times \frac{j2\pi e^{-j\pi N f_o \tau}}{j2\pi e^{-j\pi f_o \tau}} \right] \\ &= \eta N \frac{\sin(\pi N f_o \tau)}{\sin(\pi f_o \tau)} e^{-j\pi(N-1) f_o \tau}. \end{aligned}$$

An implementation of the matched-filter detector is shown in Fig. 4.1, where the filter coeffcient $p_i = p^*(\frac{iT_b}{\eta N})$. The received signal $r(t)$ is sampled ηN times per T_b. The sampled values enter the matched filter and the peaks in the output waveform correspond to the peaks in the autocorrelation function $a(\tau)$ around $\tau = 0$. The sampled values are

$$\begin{aligned} r(\frac{iT_b}{\eta N}) &= s(\frac{iT_b}{\eta N}) + n(\frac{iT_b}{\eta N}) \\ &= a_k p(\frac{iT_b}{\eta N} + \tau) + n(\frac{iT_b}{\eta N}). \end{aligned}$$

Fig. 4.2, 4.3, 4.4, 4.5 show the matched filter output with $N = 64$, corresponding to a sequence of information bits $\{a_k\} = \{..., +, -, +, +, ...\}$. Note that the off-synch spikes between like information bits, e.g. $\{..., +, +, ...\}$, are generally small, while those between opposite information bits, e.g. $\{..., +, -, ...\}$ or $\{..., -, +, ...\}$, can be larger.

The data shaping comparator performs the detection. It employs certain, possibly ad hoc, methods to look for signals only at or near the correctly synchronized samples. We assume a certain synchronization preamble is used to

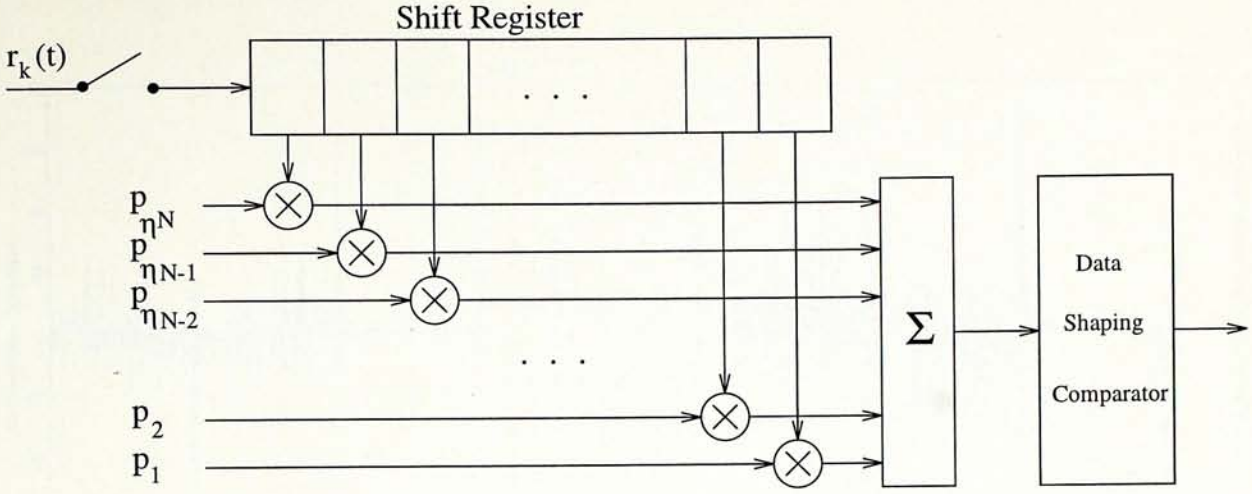


Figure 4.1: The Block Diagram of the Matched Filter

assist the comparator determine the bit (frame) synchronization of the received waveform, i.e. to determine which sample within each frame of ηN samples to perform detection. Then a binary decision on that sample produces the detector output $\hat{a}_k$. Matched filter output samples between detection moments are ignored.

There are many known implementations of the synchronization preamble. For example, a string of +'s can be transmitted (periodically if necessary), resulting in a signal which produces distinct peaks at the matched filter output. The comparator can also inspect more than one samples in the vicinity of the synchronized moment in order to make a better decision on $\hat{a}_k$. The detection moments can also be adaptively adjusted to improve performance. However, the sample timing of $r(t)$ is not adjusted. We use a fixed clock to trigger the analog-to-digital converter.

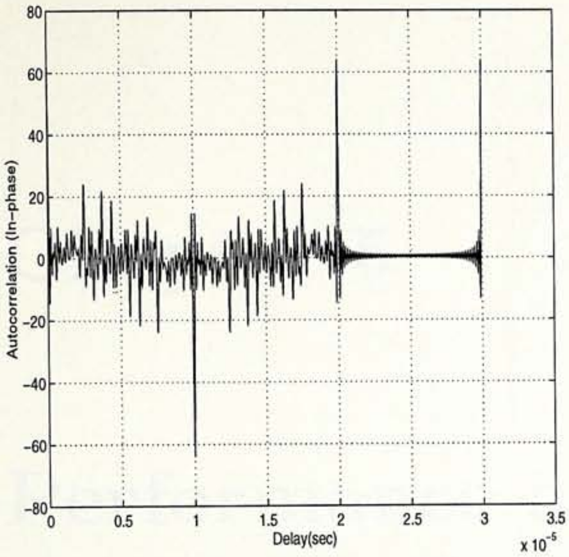


Figure 4.2: Autocorrelation for Complex modulated Signal (In-phase Part)

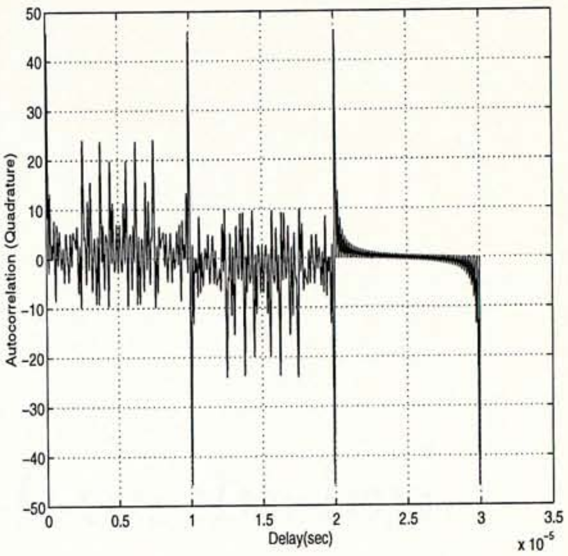


Figure 4.3: Autocorrelation for Complex modulated Signal (Quadrature Part)

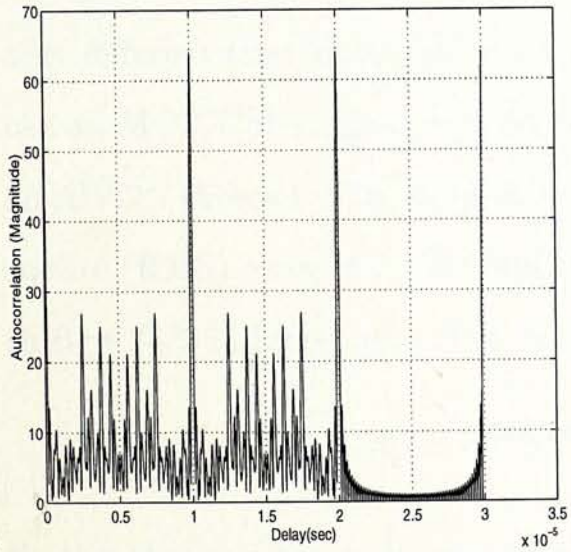


Figure 4.4: Autocorrelation for Complex modulated Signal (Magnitude)

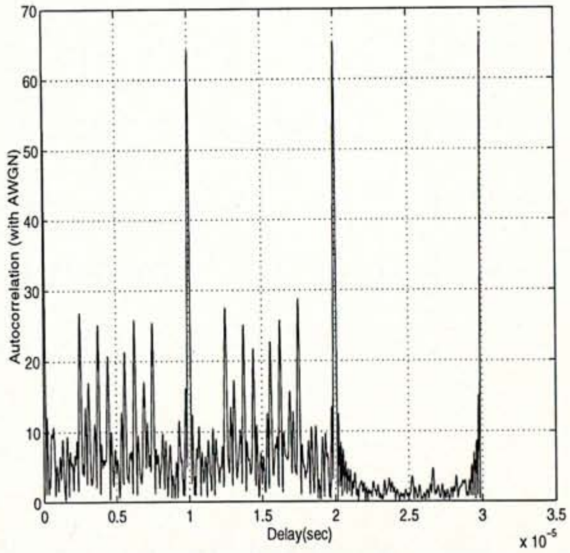


Figure 4.5: Autocorrelation for Complex modulated Signal (with AWGN)

Chapter 5

Performance and Conclusion

5.1 Line Fitting Algorithm

The performance of the phase estimator for unmodulated signal was evaluated over different time shifts and signal-to-noise ratio (SNR). In the simulation, we use an MC-CDMA signal with 64 carriers and carrier frequency $f_o = 10kHz$ in an AWGN channel. The number of iterations is 10. The result is the root-mean-square (RMS) error of $\tilde{\tau}$. We find that the performance is close except for time shift $= T_b/2$ in large noise (Fig. 5.1). When $\tau = T_b/2$ and $\theta = 0$,

$$R_n = e^{j2\pi n f_o (T_b/2)} + \epsilon_n = e^{jn\pi} + \epsilon_n.$$

So the phase vector in absence of noise is $(0, \pi, \cdots, 0, \pi)$. For those values $= \pi$, noise will easily change them to negative values and cycle compensation is hard to apply here.

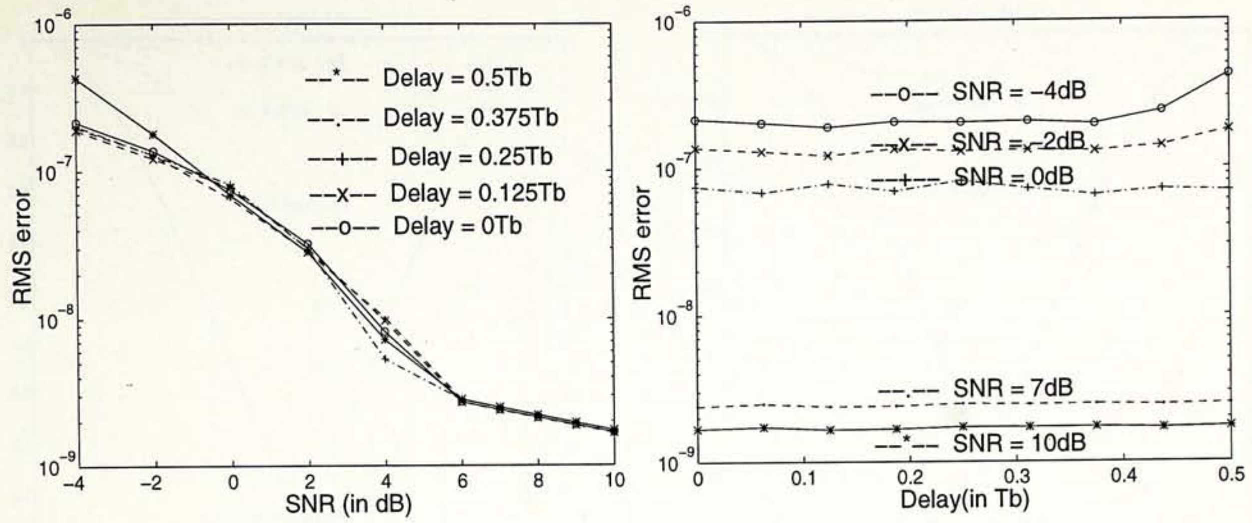


Figure 5.1: The RMS Error against SNR(left) and Time Shift (right) for unmodulated model

In practical MC-CDMA system, we can use the unmodulated synchronization preamble for initial time shift acquisition. It is similar to use many iterations in the simulation to improve the performance.

5.2 Matched Filter

Assume the noise values $n_i = n(iT_b/\eta N)$ are i.i.d Gaussian random variables with zero mean and variance σ^2. Then the signal-to-noise ratio of the transveral matched filter is [12]

$$\begin{aligned} SNR &= |\sum_{i=1}^{\eta N} p_i s(iT_b/\eta N)|^2/(\sigma^2 \sum_{i=1}^{\eta N} |p_i|^2) \\ &= |a(\tau)|^2/(\sigma^2 a(0)) \\ &= \eta |\sum_{n=0}^{N-1} e^{j2\pi n f_o \tau}|^2/(\sigma^2) \\ &= \frac{\eta}{\sigma^2}\left[\frac{\sin^2(\pi N f_o \tau)}{\sin^2(\pi f_o \tau)}\right]. \end{aligned}$$

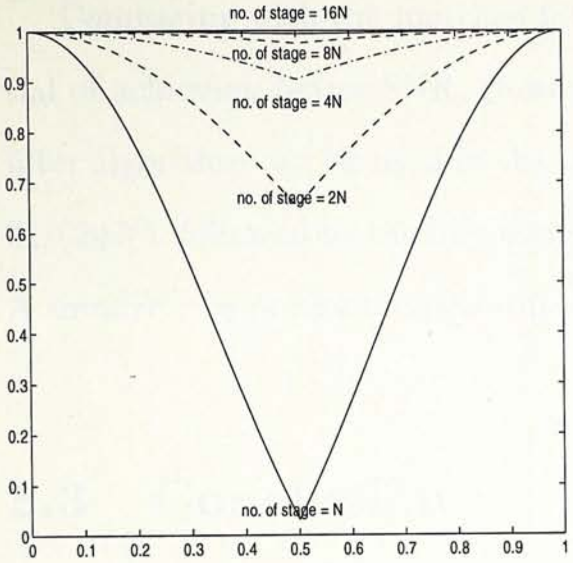


Figure 5.2: The Minimum Peak for different η for in-phase modulation

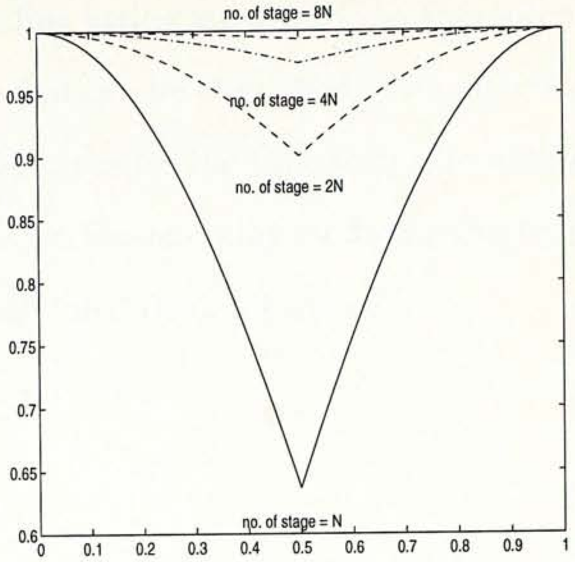


Figure 5.3: The Minimum Peak for different η for complex modulation

Assume an intelligent algorithm is used to perform symbol detections on the filter output samples nearest to the peaks, then $|\tau| \leq T_b/(2\eta N)$. However, no feedback mechanism, such as a PLL, is used to reduce τ further. We anslyze the relationship between SNR and τ in Figure 5.3 with $N = 64$ (horizontally $1 = T_b/(\eta N)$ and vertically $1{=}\sqrt{SNR}$ at $\tau = 0$). The performances can degrade as much as 3.9 dB at $\tau = T_b/(2\eta N)$, when $\eta = 1$. Increases η, i.e. sampling faster reduces the degradation, but also increases the complexity and hence it is expensive.

Remark: (1) We use I/Q demodulations above. If only in-phase modulation and demodulation are used, then the performance slack-off due to off-peak sampling is more severe (Figure 5.2, and with different vertical scale). (2) Increasing η also increases the noise power σ^2. Usually there is no net increase in SNR at $\tau = 0$.

Comparing with the matched filter, the line fitting algorithm has the potential of achieving better SNR. However, the two can be combined. The matched filter algorithm can be used in the preamble to reduce the time shift τ to within $T_b/(2\eta N)$, followed by the line-fitting algorithm for detecting modulated signals. A smaller τ improves the performance of the line fitting algorithm.

5.3 Conclusion

In the first part of the thesis, we proposed two algorithms for detecting multi-carrier CDMA signals. A fixed sampling clock is used to keep the circuitry simple. No feedback mechanism such as a PLL is used. The matched filter calculates the correlation of the received signal with the pulse. The peak value of the function can determine the time shift of the signal. The performance of the matched filter is limited by the number of stage ηN (or the sampling rate). The line fitting algorithm uses the linear property of the phase spectrum of the multi-carrier signal. It has potentially higher performance, but further research is needed to reduce its complexity.

Part II

Security on Internet

Chapter 6

Introduction

6.1 Introduction to Cryptography

Cryptography has its history for over thousands of years. In the past, it is mainly for military use to enable two parties to exchange secret without being known by the third party. Before the invention of computing machine or computer, the process to convert the message is done by human or just some simple equipment. The studies of these methods are called **Classical Cryptography**.

A simple cryptosystem is shown in Figure 6.1, the user **A** who sends the message is called the **sender** and the user **B** who receives the message is called the **receiver**. The message sent by **A** is called **plaintext** and the process to convert the plaintext into some messages which cannot be understood by the third party is called **encryption**. The message after encryption is called **ciphertext** and the reverse process to convert the ciphertext back to its original

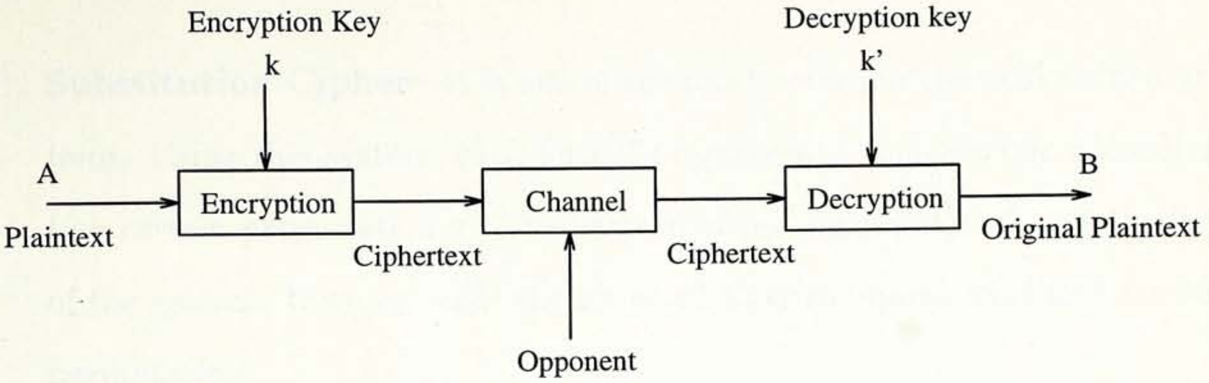


Figure 6.1: Simple Encryption and Decryption Model

plaintext is called **decryption**. The media for the transmission of the ciphertext is called **channel** which can be human, letter, telephone, radio or computer. In most cases, the encryption and the decryption processes require some parameters called **key**. The key for encryption is **encryption key k** whereas the key for decryption is **decryption key k'** [20].

In modern technology, the advance of computer and telecommunication networks introduces many new problems that require some advanced cryptographic algorithms to solve them. These problems includes (i) verification of user identity, (ii) transmission of secret personal data, (iii) transmission of electronic signed document, (iv) electronic commerce and so on.

6.1.1 Classical Cryptography

Before we talk about the cryptosystems used nowadays, lets first have an overview of some classical cryptosystems used in past. The main idea beyond these cryptosystems is substitution and transposition, which means that to replace certain letters with others or to rearrange the order of the letters in the message [19, 20].

1. **Substitution Cipher** - it is one of the most common classical cryptosystems. Using this system, each letter is replaced by another one according to a certain permutation π. The pattern of the permutation forms the key of the system. If we consider the set of all English alphabets, there are 26! permutation.

2. **Shift Cipher** - it is the simplest form of substitution cipher. The permutation is just a fixed shift of k, i.e. each alphabet is shifted by k. If the shift $k = 3$, it is called the **Caesar Cipher**, which was used by Julius Caesar. In this case, a is shifted to D, b is shifted to E and so on.

3. **The *Vigenère* Cipher** - the cryptosystems mentioned above is called a monoalphabetic substitution, each letter is replaced by another one. In *Vigenère* Cipher, each time m letters are encrypted by the key $k = \{k_1, k_2, \cdots, k_m\}$. For the plaintext $\mathcal{P} = x_1, x_2, \cdots$, the ciphertext will be $\mathcal{C} = y_1, y_2, \cdots$ where $y_{jm+i} = x_{jm+i} + k_i \pmod{26} \quad i = 0, 1, \cdots, m-1$. The *Vigenère* Cipher is a kind of polyalphabetic substitution.

4. **Transposition Cipher** - for the **Substitution Cipher** discussed above, the plaintext is replaced by some other characters. In **Transposition Cipher** (or sometimes called **Permutation Cipher**), all the characters in the plaintext remain the same, except that the order of them are rearranged. Suppose we have a key with length m, then the plaintext is divided into many groups of characters of length m. For each group of the characters, they are permutated according to the pattern of the key.

6.1.2 Cryptanalysis

Cryptanalysis is the study or science to determine the original plaintext without the knowledge of the key. There is a general assumption that the type cryptosystem used is known. People who study cryptanalysis is known as **Cryptanalysts**. With some knowledge of the plaintext or ciphertext, cryptanalyst can increase the chance of successful cryptanalysis. Some general type of cryptanalysis are shown as follow in order of their strength, (i) ciphertext-only, (ii) known plaintext, (iii) chosen plaintext and (iv) chosen ciphertext.

6.2 Introduction to Internet Security

6.2.1 The Origin of Internet

In early years, most networks are operated independently by a small group of people. And it is almost impossible to setup a global network to cover all users. The evolution of internetworking technology makes it possible to interconnect different physical networks over the world transparently. One main early Internet technology are developed by the **Defense Advanced Research Projects Agency** (DARPA) in mid 1970s. They developed the standard for internetworking and routing which is called **Transmission Control Protocol/Internet Protocol** (TCP/IP) [21]. The TCP/IP technology was employed to connect many research institutions to those of DARPA and is so called **connected Internet** or just **Internet**.

In around 1980, the ARPANET was used as the backbone of the new Internet using TCP/IP technology. In early 1983, the Defense Communication Agency (DCA) splitted the ARPANET into the research part which still called ARPANET and the military part called MILNET. The ARPANET technology has great influence to today's Internet. Some early Internet services include [N] :

1. Electronic mail - Simple Mail Transfer Protocol (SMTP).
2. File transfer - File Transfer Protocol (FTP) and Network File System (NFS).
3. Remote login - TELNET, to connect remote systems via the network.
4. Information services - Gopher, Wide Area Information Service (WAIS) and World Wide Web (WWW).
5. Remote Procedure Call (RPC) services - Network File System (NFS), allows systems to share directories and disks and Network Information Services (NIS), allows multiple systems to share database.

6.2.2 Internet Security

The first thing related to network and Internet security is password. A user needs to enter his password while login so as to identify his identity. Therefore, a proper selection of password is very important. Within a local network, there are many ways to enforce security. The security problem become more complex if we consider interconnected network. The rapid growth and openness of Internet make Internet vulnerable and a easy path to attack connected hosts. Moreover, these security problems become a barrier for people and companies who want to join or do business in the Internet.

Today, private key cryptosystems are widely employed in conjunction with public key cryptosystems to enhance Internet security. Private key cryptosystems are used for messsage encryption and decryption and public key cryptosystems are used for key distribution and authentication. Authentication is achieved by means of digital signature based on public key cryptography which is analogous to written signature. Moreover, one-way hashing or message digest function like Message Digest 5 (MD5) and Secure Hash Algorithm (SHA) can provide message integrity check.

Meanwhile, it is also common to setup firewalls for Internet security. A firewall acts as a filter between the Internet and the trusted network such that only authorized information can pass through the network. Moreover, many applications are now available for Internet security like Kerberos for authentication, Privacy-Enhanced Mail (PEM) and Pretty Good Privacy (PGP) for electronic mail and message encryption. Besides, many Internet browsers nowaday also apply different private and public key algorithms to provide their security services like Secure NCSA Mosaic, Netscape Navigator and SunSoft HotJava.

6.2.3 Internet Commerce

As mentioned before, there is rapid increase in the number of Internet subscribers. Moreover, many different services for Internet are now available, they include newsgroup, ftp, gopher and many be most important, e-mail and World-wide Web. Through these services, people can access information all over the world without leaving homes or offices. For example, a user can communicate with other users overseas using e-mail, newsgroup and Internet phone. A user can also download some softwares, documents by FTP. With the use of World-wide Web, users can access many Internet services and data with different formats like image, audio and video.

The rapid growth, wide range of provided services and popularity of Internet initiates a new environment and media for merchant to do business. Through the use of Internet, merchants can promote their products and services by different media like text, image, audio and video. Meanwhile, a consumer can send his feedback or purchase order to the merchant through Internet.

To purchase over the Internet, a consumer can select the desired products or services from the merchant's homepage and pay the money offline. However, to increase flexibility, it is desirable that the consumer can pay directly over the Internet. For example, the payer can send his credit card number over the Internet through e-mail or other formats and the payee can then charge the payer's account. However, the Internet is vulnerable to attack, so it is important to design a system to ensure secure transaction over the Internet. Many different electronic commerce systems have been proposed and in use now, a consumer can pay using his credit card or two new payment methods, electronic cheque and electronic cash. These payment methods deploy many advance private and public key cryptographic algorithms for encryption and authentication.

Chapter 7

Elementary Number Theory

7.1 Finite Field Theory

In modern cryptography, it is common to perform computation over finite field in many popular private and public key algorithms (eg. IDEA, RC5, RSA, Diffie-Hellman). So lets first have an overview of finite field theory and some related algorithms.

The ring formed by the residue elements of the integer ring over modulo q operation is called a quotient ring $\mathcal{Z}/(q)$ with elements $\{0, 1, 2, \cdots, q-1\}$. The quotient ring $\mathcal{Z}/(p)$ formed by the residue elements of the integer ring modulo a prime number p is a finite field. A finite field can also be considered as a field that has finite number of elements and the number of elements in a finite field is called the order. The field $\mathcal{Z}/(q)$ is an example of finite field and called the **Galois field** $(GF(q))$.

7.1.1 Euclidean Algorithm

The **Euclidean Algorithm** is used to compute the **Greatest Common Divisor** (GCD) of two integers. For any integers a, b, r and q, if $a = qb + r$ then, $gcd(a, b) = gcd(b, r)$. For any two integer a, p, if $gcd(a, p) = 1$, then they are said to be relatively prime or coprime.

Fermat's Little Theorem

If p is a prime, and p does not divide a, then $a^{p-1} = 1 \pmod p$.

Extended Euclidean Algorithm

The **Extended Euclidean Algorithm** is used to find the integer solutions x and y satisfying the equation $ax + py = gcd(a, p)$. If we consider the equation in modulo p, then $ax = gcd(a, p) \pmod p$. When $gcd(a, p) = 1$, the solution x is said to be the inverse of a modulo p, $a^{-1} \equiv x \pmod p$ [19].

7.1.2 Chinese Remainder Theorem

The **Chinese Remainder Theorem** was first proposed by a great Chinese military strategist Suen Zi. He used this theorem to count the number of soldiers in the army [19]. This problem can be formulated as a system of equations and solve x,

$$\begin{cases} x & \equiv & a_1 \pmod{p_1} \\ x & \equiv & a_2 \pmod{p_2} \\ \multicolumn{3}{c}{\cdots\cdots\cdots\cdots\cdots\cdots} \\ x & \equiv & a_n \pmod{p_n}, \end{cases}$$

where $p_1, p_2, \cdots, p_n$ are all positive integers with $gcd(p_i, p_j) = 1$ and $i \neq j$. Then we can find the solution x satisfys the equation $x \equiv a \pmod{P}$ with $P = \prod_{i=1}^{n} p_i$. Let $P_i = P/p_i$ (for $1 \leq i \leq n$), then use the Extended Euclidean Algorithm to find the inverse of P_i modulo p_i, that is $q_i = P_i^{-1} \pmod{p_i}$ (for $1 \leq i \leq n$). The solution will be,

$$x = \sum_{i=1}^{n} q_i P_i a_i \pmod{P}.$$

Some examples of Chinese Remainder Theorem are described in the Appendix.

7.1.3 Modular Exponentiation

In many public-key cryptosystems (like RSA and ElGamal), the encryption and decryption processes must involve the computation of the power of an integer modulo p ($a^b \pmod{p}$). This process is known as **modular exponentiation** [19]. One simple calculation is the **square and multiply** method. To calculate $a^b \pmod{p}$, first find the binary representation of b such that $b = \sum_{i=0}^{k-1} b_i 2^i$. Then first initialize with $y = 1$ and start from $i = k - 1$ and down to 0,

1. Replace $y = y^2 \pmod{p}$.

2. If $b_i = 1$, replace $y = y \times a \pmod{p}$.

The resultant value will be $y = a^b \pmod{p}$. The main idea is to reduce the problem to modular multiplication so that at most $2k$ **modular multiplications** are required. Furthermore, if p is a prime number and does not divide a, applying Fermat's Little Theorem that $a^{p-1} = 1 \pmod{p}$, the modular exponentiation can be simplified by $a^b \pmod{p} = a^c \pmod{p}$ where $c = b \pmod{p-1}$. Then use the algorithm above with power c such that the number of multiplication can be further reduced.

7.2 One-way Hashing Function

1. An **One-way Hashing Function** h is defined as, for an arbitrary message M, it is easy to compute its **message digest** $d = h(M)$. But it is computationally infeasible to compute the message M from d.

2. For an arbitrary message M, if it is computationally infeasible to find another message M' such that $h(M) = h(M')$, then h is **weakly collision-free**.

3. If it is computationally infeasible to find two messages M and M' such that $h(M) = h(M')$, then h is **strongly collision-free** [19].

According to *birthday paradox*, you need 183 person such that the probability that one of them has the same birthday as you is greater than 1/2. However, only 23 person is needed such that the probability that there exists two person with same birthday is greater than 1/2. This phenomenon is also known as the *birthday attack*.

The problem of birthday attack is related to the size of the message digest d. If the message digest is m-bit long, it needs 2^m random trials to find a message which has same message digest as a specific message. However, like birthday attack, it needs only $2^{m/2}$ random trials to find two messages which have same message digest. It is suggested that the length of the message digest should be at least 128 bits, so the birthday attack needs 2^{64} random trials to find two messages with same message digest.

7.2.1 MD2

MD2 is a message-digest algorithm proposed by J. Linn and R. Rivest. It is an algorithm to perform one-way hashing of an input message of arbitrary length and outputs a 128-bit *message digest*. This algorithm [A] is designed for compressing a document before signed by digital signature of a public-key cryptosystem. It is used by PEM, an Internet Privacy Enhanced Mail standard.

7.2.2 MD5

MD5 is another message-digest algorithm proposed by R. Rivest [B]. Like MD2, it is an algorithm to perform one-way hashing of an input message of arbitrary length and outputs a 128-bit message digest.

Firstly, padding bits are appended to the message so that the length will be 448 mod 512. Then the 64-bit representation of the message length before padding is appended to the padded message. In this way, the length of the resultant message will be a multiple of 512 bits. For each iteration of the main loop, a block of 512 bits (sixteen 32-bit *words*) padded message is used for manipulation. In the next step, 4 registers are initialized with A = 0x67452301, B = 0xefcdab89, C = 0x98badcfe and D = 0x10325476.

These initial values are stored as AA, BB, CC and DD, and A, B, C and D will then be used in the main loop. The main loop has 4 rounds and each round has 16 operations. Each operation first involves a function that intakes three of the registers (A,B,C,D) and performs bitwise operation (*AND*, *OR*, *NOT*, *XOR*) to them. The output is added to the remained register, one of the sixteen 32-bit block of the message and a special constant t. The output sum will undergo a circular left shift of s bits and finally add this result back to one of the register.

After these 64 operations, the resultant values A, B, C and D are added back to their initial values AA, BB, CC and DD. The 128-bit message digest will be the concatenation of these results A, B, C and D.

7.3 Prime Number

A **Prime number** is a number which has no factor except 1 and itself. A number which has factor other than 1 and itself is **composite**. Prime number is used in generation of key in some public key cryptosystems like Diffie-Hellman and RSA. Many proofs have already shown that there are infinite many primes and the number of primes $< x$ is approximately $x/\log x$ [23, 24].

7.3.1 Listing of Prime Number

Two simple ways to find a list of prime numbers includes [24],

1. **Trial by Division** - For an arbitrary integer n, if there does not exist any integer $p \leq \sqrt{n}$ which is divisible by n, then n is a prime number.
2. **Sieve of Eratosthenes** - This method relys on the elimination of composite. Firstly, 2 is sieved so as to remove all multiples of 2 except 2. Then the smallest unsieved number left, 3 is sieved. The process is repeated for $5, 7, 11, \cdots, p_k, p_{k+1}$ so that all composite $< p_{k+1}^2$ will be eliminated.

7.3.2 Primality Testing

The prime numbers used in cryptosystem are usually very large (up to several hundreds of bits). To generate such a large prime, it is common to generate a large random number and test for primality. Those numbers who pass the **primality testing** have high probabilities that they are prime. Common algorithms include the **Solovay and Strassen's test** and the **Rabin's test** discussed below.

Quadratic Residues

Consider a prime $p > 2$ that does not divides an integer a. If the congruence $a \equiv b^2 \pmod{p}$ has a solution $b \in \mathcal{Z}_p$, then a is said to be a **quadratic residue modulo p**, otherwise, a is a **nonquadratic residue modulo p** [19, 23].

From this, Legendre has defined the **Legendre symbol**,

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & a \equiv 0 \pmod{p} \\ 1 & \text{a is a quadratic residue modulo p} \\ -1 & \text{a is a nonquadratic residue modulo p.} \end{cases}$$

Solovay and Strassen's Test

Solovay and Strassen's Method is a *Monte Carlo* primality testing method based on the properties of Legendre symbol [23]. To test if an integer N is prime,

1. Choose an integer $a < N$ such that $gcd(a, N) = 1$.
2. Test if the *Euler's Criterion* is satisified,

$$\left(\frac{a}{N}\right) \equiv a^{(N-1)/2} \pmod{N}.$$

If the above congruence does not hold, then N is said to be composite. If it holds, repeats step 1 and 2. If the test passed for k times, then N is a prime with a probability of at least $1 - 1/2^k$.

Rabin's Test

This method is based on the Miller's test which is more accurate than Solovay and Strassen's test [23]. To test if an integer N is prime,

1. Choose an integer $a < N$ such that $gcd(a, N) = 1$.

2. Test if N satisfies the condition for strong pseudoprime in base a,

$$\begin{aligned} a^d &\equiv 1 \pmod N && d \text{ is odd such that } N-1=2^s d \\ a^{2^r d} &\equiv -1 \pmod N && \text{for some } r, 0 \le r \le s. \end{aligned}$$

If the above congruences do not hold, then N is said to be composite. If they hold, repeats step 1 and 2. If the test passed for k times, then N is a prime with a probability of at least $1 - 1/4^k$.

7.4 Random/Pseudo-Random Number

Random numbers (RN) are widely used in many cryptosystems. For example, to generate cryptographic keys or passwords, random numbers are needed [C, 27]. The selection of random source or random-number generating technique has great influence to the security of a cryptosystem.

In many computer applications, random numbers are generated by some deterministic algorithms. These numbers generated are not truly random, in fact, they should be denoted as **pseudo-random numbers** (PRN). A desired PRN sequence should have, (i) sufficently large least period, (ii) good statistical properties and (iii) efficient algorithms [26].

Examples of Hardware Random Source

The first example is use the input of some analog audio or video source to a computer. The digitized input source can be an audio digitizer with no input source or a video camera with the lens covered such that the detected input will like thermal noise. The second method uses low level disk seek time of a system with small random fluctuations in disk drives rotational speed because of chaotic air turbulence [27]. Other examples include radioactive source, quantum effects in semiconductor devices [C], measure of timing and content of mouse movement, key strokes under some requirements [27].

Insecure Random Source

Some random sources or generating techniques which are commonly used in many general purpose applications may be insecure for cryptograhical use. Examples include the system clock value, process runtime and other network statistics. Moreover, the use of a very complex random number generation algorithm may not help if there is no theory behind or analysis of the algorithm. Published data on CD or other large database are also insecure if the adversary can also access the same database [C, 27].

Uses of Mixing Function

One way to increase the randomness of a number is to use a number of correlated random input sources and mix them together by mixing functions like hashing or message digest functions [27]. These functions use inputs of arbitrary length and output a fixed length output (160 bits for SHA and 128 bits for MD2, MD4, MD5). As described before, it is difficult to compute the input from the message digest. Besides message digest functions, some encryption algorithms can also be used for mixing function like DES and Diffie-Hellman.

7.4.1 Examples of Random Number Generator

Linear Congruential Generator

First select integers a, b, n such that $n \geq 2$ and $1 \leq a, b \leq n-1$. Define $k = \lceil \log_2 n \rceil$ and let $k+1 \leq l \leq n-1$. For a seed s_0, where $0 \leq s_0 \leq n-1$, start from $i = 1$ to l,

$$s_i = (as_{i-1} + b) \pmod{n},$$

and the l-bit generator output is $r = (r_1, r_2, \ldots, r_l)$ where r_i is the least significant bit of s_i. This generator has a period no greater than n. If a, b, and n are properly chosen, then the generator will be a *maximal length generator* with period n. However, some studies show that linear congruential generator is not secure enough for cryptographical applications [18].

RSA Generator

Let p, q to be two $k/2$-bit primes, and define $n = pq$. Select b such that $gcd(b, \phi(n)) = 1$. Like RSA cryptosystem, n and b are public while p and q are kept secret. The seed s_0 is any k-bit element of $\mathcal{Z}_n^*$. Start form $i = 1$ to l, define

$$s_{i+1} = s_i^b \pmod{n},$$

and the generator output is $r = (r_1, r_2, \ldots, r_l)$ where r_i is the least significant bit of s_i. The security of RSA generator depends on the difficulty to break RSA cryptosystem [19].

Blum-Blum-Shub Generator

This generator is based on the quadratic residue modulo n and Legendre symbol defined before [18, 19]. First select two primes p, q where $p \equiv q \equiv 3 \pmod 4$ and $n = pq$. The seed s_0 used should be a quadratic residues modulo n ($s_0 \in \{x^2 \pmod n : x \in \mathcal{Z}_n^*\}$). Start from $i = 1$ to l, define,

$$s_i = s_{i-1}^2 \pmod n,$$

and the generator output is $r = (r_1, r_2, \ldots, r_l)$ where r_i is the least significant $\log_2(\log_2(s_i))$ bits of s_i. Moreover, it is possible to compute a particular s_i directly from s_0 since,

$$\begin{aligned} s_i &= s_{i-1}^2 \pmod n = (s_{i-2}^2 \pmod n)^2 \pmod n = \cdots \\ &= s_0^{2^i} \pmod n = s_0^{2^i \pmod{\phi(n)}} \pmod n, \end{aligned}$$

where $\phi(n) = (p-1)(q-1)$. Just like the RSA generator, n is public while p and q are kept secret. The security of BBS generator is rely on the difficult to factor n.

Chapter 8

Private Key and Public Key Cryptography

8.1 Block Ciphers

In **block cipher**, each time a fixed length of plaintext is converted to a fixed length ciphertext. The design of the encryption and decryption process is that every bit in a ciphertext block should jointly depends on every bit in the plaintext block and key block. A block of random number called the *initialization vector* is used for the encryption of the first plaintext block so that the same plaintext will not encrypt to same ciphertext each time.

Since the size of the message may not be multiple of block size, certain padding bit string should be appended to the end of the message. The bit string can be a sequence of zero with a *pad count* at the end to specify the number of bits appended. If the message is a multiple of block size, an extra block of padding should be appended to the message [28]. Using **cipher block chaining** (CBC) mode, each time the plaintext block is XORed with the previous ciphertext block before encryption. Using **cipher feedback** (CFB) mode, each time a data portion smaller than the block size is encrypted [18].

8.1.1 Data Encryption Standard (DES)

In early 1970s' the National Bureau of Standards (NBS), now the National Institute of Standards and Technology (NIST), started a program to request the US industry to develop a standard cryptosystem. One proposal is the Lucifer developed by IBM, after some modifications and simplifications, it became the Data Encryption Standard (DES) in 1977 [18, 29]. DES is a block cipher with 64 bits block size and keysize. However, only 56 bits of the key is used whereas the remaining 8 bits are used for parity checks.

The encryption process is shown in Figure 8.1, the message is first divided into blocks of 64 bits size. Each 64-bit block is permuted by the *initial permutation* (IP) such that the 58-th bit goes to output bit 1, 50-th bit goes to bit 2 and so on. Then let the leftmost 32 bits be L_0 and the rightmost 32 bits be R_0. They become the input of 16-round iterations. Suppose L_{i-1} and $R_{i-1} (1 \leq i \leq 16)$ be the leftmost 32 bits and the rightmost 32 bits respectively in each round. Then,

$$L_i = R_{i-1} \qquad \text{and} \qquad R_i = L_{i-1} \oplus f(R_{i-1}, K_i).$$

The 48-bit K_i is generated by the 64-bit encryption key K. In K, bits $8, 16, \cdots 64$ are the odd parity check bits. The remaining 56 bits are extracted

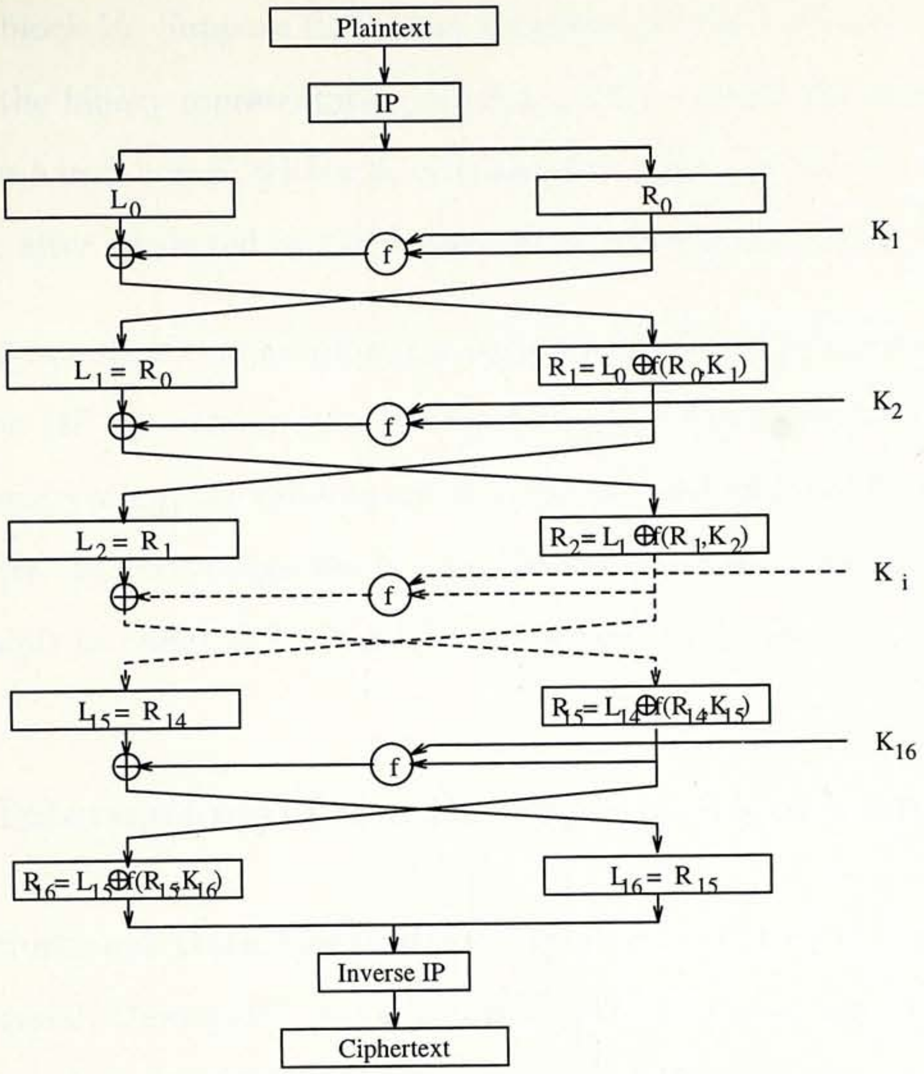


Figure 8.1: The Encryption Algorithm of DES

and permutated by the *key permutation* (PC-1). Let the leftmost 28-bit output of PC-1 be K' and the rightmost 28-bit be K'', in each of the subsequent round, K' and K'' will undergo a cyclic leftshift of 1 or 2 bits. The shift in rounds 1, 2, 9 and 16 is 1 where the other rounds are 2. The shifted output in round i will be permuted by the *compression permutation* (PC-2) and the output is K_i.

For the function f, first the 32-bit R_i is expanded to a 48-bit block by a *expansion permutation* (EP), bitwisely XORed with K_i and then divided into eight 6-bit subblock $U_j (1 \leq j \leq 8)$. Each U_j will enter the *S-box* S_j and outputs a 4-bit subblock V_j. Suppose the binary representation of U_i is $u_1u_2\cdots u_6$, then V_j will be the binary representation of $(S_j)_{ab}$. $(S_j)_{ab}$ means the entry of row a and column b in S-box S_j with $a = u_1u_6$ and $b = u_2u_3u_4u_5$. The S-box outputs $U_1U_2\cdots U_8$ after permuted by the *P-box* will be the output $f(R_{i-1}, K_i)$.

After 16 rounds of computation, the output is permuted by the *inverse initial permutation* (IP^{-1}) and produce the final output. The decryption process is similar to encryption, the same function f can be used with the key K_i used in reverse order. In decryption, the key subblock K' and K'' undergo cyclic right shift, the shift in round 1, 8, 15 and 16 is 1 where the others are 2 [18, 29].

8.1.2 International Data Encryption Algorithm (IDEA)

The **International Data Encryption Algorithm** (IDEA) is first developed by X. Lai and J. Massey [D]. It is a symmetrical block cipher algorithm with 64-bit block length and 128-bit key (twice as long as for DES). The main operation of this algorithm involves three main algebraic groups that operate on 16-bit subblock each time includes, (i) bitwise *XOR*, (ii) modulo 2^{16} addition, and (iii) modulo $2^{16} + 1$ (*Fermat prime*) multiplication with value '0' not used but uses 2^{16} to represent all-zero 16-bit block.

The encryption and decryption algorithms can be illustrated by Fig.8.2. Firstly, the 64-bit block of plaintext is divided into four 16-bit subblocks, X_1, X_2, X_3 and X_4. These four 16-bit subblocks are transformed into four 16-bit ciphertext subblocks Y_1, Y_2, Y_3 and Y_4 through 9 rounds of operation. In each of the first 8 rounds, those intermediate 16-bit subblocks will undergo XOR, addition and multiplication described above with six 16-bit keys $Z_1^i, Z_2^i, \cdots, Z_6^i (i = 1, 2, \cdots, 8)$. In the last round, only addition and multiplication are involved with four 16-bit keys Z_1^9, Z_2^9, Z_3^9 and Z_4^9. So there are total 52 key blocks used for both encryption and decryption, which is generated by the 128-bit key.

According to Ascom Systec, IDEA has been implemented as a chip with an encryption rate of 177 Mbit/s on a device with a system clock frequency of 25 MHz [30]. Moreover, this algorithm is now available to North America [E].

8.1.3 RC5

RC5 is a new variable-key-size symmetric block cipher proposed by R. L. Rivest of RSA Data Security [F]. RC5 is a parameterized algorithm and can be denoted as $RC5 - w/r/b$ with parameters w, r and b,

1. w - The word size in bits and possible values are 16, 32 and 64. The plaintext and ciphertext blocks are two-word blocks ($2w$ bits).
2. r - The number of rounds and possible values are 0 to 255. The user can select different value to trade-off between time efficiency and security.
3. b - The private key size in bytes and possible values are 0 to 255. Flexible key-size can provide different security and fulfill the export restriction.

For example, RC5-32/16/7 can provide the same number of rounds and key-size as in DES. The proposed "nominal" values of the parameters is RC5-32/12/16.

The main algorithm of RC5 consists of **encryption**, **decryption** and the **key expansion** algorithm. Three primitive operations are used in this algorithm:

1. Modulo 2^w addition of words using two's complement and denoted by "+",
2. bitwise *XOR* denoted by "$\oplus$", and
3. cyclic left rotation of words denoted by $x <<< s$.

The *key expansion* expands the private key K to an expanded key array S of size $2(r+1)$. This algorithm uses two $2w$ length binary constants P_w and Q_w,

$$P_w = Odd((e-2)2^w) \qquad \text{and} \qquad Q_w = Odd((\phi - 1)2^w).$$

$Odd(x)$ means the odd integer nearest to x, e is the base of natural logarithms and ϕ is the golden ratio. The cyclic rotation of RC5 is the only non-linear operator and the number of rotation is a variable (plaintext-dependent). The strength of RC5 greatly depends on the data-dependent rotations since the amount of rotation is not predetermined.

8.2 Stream Ciphers

In **stream cipher**, each time only one bit of plaintext is encrypted into ciphertext. The encryption process is that the *encryption key* enters the keystream generator and outputs a keystream $k_1, k_2, \cdots, k_i$. The plaintext stream is XORed with the keystream to produce the ciphertext stream, $c_i = p_i \oplus k_i$. The decryption process is similar in which the ciphertext stream is XORed with the original keystream, $c_i \oplus k_i = p_i \oplus k_i \oplus k_i = p_i$. Like block cipher, an initialization vector should be used. One possible implementation of the keystream generator is the *linear feedback shift register* (LFSR).

In *synchronous stream cipher*, if there is one bit error in the ciphertext, the synchronization will loss. It is advantage that the advesary cannot make insertion or deletion to the message without being detected. In *self-synchronizing stream cipher*, each time n ciphertext bits are feedback to the keystream generator. Therefore one bit error will propagate to at most n ciphertext bits [31].

For a good stream ciphers, (i) the period should be large (the period is the minimum value p such that $k_{i+p} = k_i$), (ii) the shortest length of a unique LFSR to produce k_i called the *linear complexity* should be large and (iii) the keystream should be a random/pseudo-random sequence [31]. Using stream ciphers, no minimum block size is required as in block ciphers. However, initailization vector is required for all stream ciphers and must be changed frequently but not necessary for block ciphers [28].

8.2.1 RC2 and RC4

RC2 and **RC4** are two variable-key-size encryption functions designed by R. L. Rivest [G]. According to RSADSI, RC2 is a variable-key-size symmetric block cipher and can be used as an alternative of DES. RSADSI claims that RC2 is approximately twice as fast as DES (with triple encryption) when implemented in software. RC4 is a variable-key-size symmetric stream cipher and RSADSI claims that RC4 is 10 or more times as fast as DES when implemented in software.

RSADSI claims that RC2 and RC4 can be more secure than DES because they can use long key sizes, but less secure when the key sizes are short. However, both RC2 and RC4 are proprietary algorithms of RSADSI and the details of the algorithms have not been published. Therefore, it is difficult to prove that if RC2 and RC4 are really more secure than DES. Though RC2 and RC4 are variable-key-size, it is common to use key sizes of 128 bits for local use in U.S. and this version cannot export. Nevertheless, an agreement between the Software Publishers Association (SPA) and the U.S. government gives RC2 and RC4 special status so that they can export if the key sizes are limited to 40 bits.

8.3 Public Key Cryptosystem

In traditional private key crytposystems, there exist some limitations [32, 29],

1. The key management and distribution problem - suppose there are n users in the system, since each pair of user share a key which is kept secret to the others, so as a total, $n(n-1)/2$ private keys are required. Moreover, before actual communication, each pair of user requires some secure channels like mail or personal contact to exchange their keys in advance.

2. The authentication problem - it is difficult to use conventional cryptosystem as an authentication system, for example, to validate the identity of a user or to verify a signed message by digital signature.

These disadvantages raised people's interest to develop a new kind of cryptosystem. In 1976, W. Diffie and M. E. Hellman published a paper "New directions in cryptography" and proposed the concept of **public key cryptosystem.** In public key cryptosystem, the encryption key e_k and decryption key d_k are distinct. The encryption key is published while the decryption key is kept secret. It is computational infeasible to compute d_k given e_k and vice versa. Suppose user A wants to send a secret message M to user B, A will look up $B's$ encryption key in some public directories. A encrypts the message to $C = E_B(M)$ and sends it to B. When B received the ciphertext C, he decrypts C by his decryption key, $D_B(C) = D_B(E_B(M)) = M$ [32].

Trapdoor One-way Functions

The encryption and decryption algorithms employ the concept of **trapdoor one-way functions**. A function $f : \mathcal{M} \rightarrow \mathcal{C}$ is said to be **one-way** if it is computable in polynomial time to compute $f(m), m \in \mathcal{M}$ but computation infeasible to compute the inverse $f^{-1}(c), c \in \mathcal{C}$. Moreover, an one-way function $g : \mathcal{M} \rightarrow \mathcal{C}$ is said to be a **trapdoor one-way function** if a special parameter k called the **trapdoor** can make the computation of $g_k^{-1}(c), c \in \mathcal{C}$ easy [36].

A public key cryptosystem can use certain trapdoor one-way function f_e as the encryption function and the inverse f_d^{-1} as the decryption function. The parameter e is the public key and the trapdoor d is the private key. Some popular public key cryptosystems include : (i) the **Diffie-Hellman** system, (ii) the **RSA** system and (iii) the **elliptic curve** cryptosystem.

8.3.1 Diffie-Hellman

The **Diffie-Hellman** system is a public key distribution system developed by W. Diffie and M. E. Hellman [32]. Suppose users A and B want to exchanged a private key over an insecure channel before communication, they first agree on a large prime number n and a large primitive element $\alpha \in \mathcal{Z}_n$. Users A and B independently generate random numbers x and y respectively such that $1 \leq x, y \leq n-1$ and keep them secret. Then A and B compute,

$$X = \alpha^x \pmod{n} \qquad \text{and} \qquad Y = \alpha^y \pmod{n}$$

respectively and place X and Y in a public file. A lookups Y from the public file and computes the secret key K_{AB},

$$K_{AB} = Y^x \pmod{n} = \alpha^{xy} \pmod{n},$$

and B can also compute this key using X from the public file,

$$K_{AB} = X^y \pmod{n} = \alpha^{xy} \pmod{n}.$$

The security of this system depends on the difficulty to compute logarithms modulo n. It means that it is easy to calculate $X = \alpha^x \pmod{n}$ but very difficult to calculate $x = \log_\alpha X \pmod{n}$.

8.3.2 Knapsack Algorithm

The **Knapsack Algorithm** is proposed by R. Merkle and M. E. Hellman in 1978. This algorithm is based on the **knapsack problem** that given a knapsack of length S and a set $\{a\}$ with n rods of the length $a_1, a_2, \cdots, a_n$ respectively. The problem is to find a subset $\{a_s\} \subseteq \{a\}$ such that,

$$S = \sum_{a_i \in \{a_s\}} a_i = \sum_{i=1}^{n} b_i a_i,$$

where $(b_1, b_2, \cdots, b_n) = b$ is a binary sequence. The difficulty of the knapsack problem is that it is computationally infeasible to find the sequence b given the knapsack S and $\{a\}$.

The main idea of the algorithm is based on an *easy knapsack* and a *hard knapsack* problem. The public key $\{a'\}$ is the hard knapsack whereas the private keys $\{a\}, p, m$ is the easy knapsack. The knapsack is said easy if $\{a\}$ is a superincreasing sequence,

$$a_j > \sum_{i=1}^{j-1} a_i.$$

To generate the public key, we can first start from the private key $\{a\}$. The public key (hard knapsack) $\{a'\}$ will be

$$a'_i = a_i \times p \pmod{m} \qquad\qquad 1 \leq i \leq n,$$

where $m > \sum_{i=1}^{n} a_i$ and $gcd(p, m) = 1$ $(1 \leq i \leq n)$.

To encrypt a message with the public key $\{a'\}$, $\{b\}$ will be the message block, so that the value of b_i will decide whether a'_i is selected and the ciphertext is S. To decrypt the message, compute

$$\begin{aligned} S' &= p^{-1} \times S \pmod{m} = p^{-1} \times \sum_{i=1}^{n} b_i a'_i \pmod{m} \\ &= p^{-1} \times \sum_{i=1}^{n} b_i (a_i \times p) \pmod{m} = \sum_{i=1}^{n} b_i a_i \pmod{m}, \end{aligned}$$

and the solution of these easy knapsack S' will be the plaintext. To solve the easy knapsack given S' and $\{a\}$, first compare a_n and S'. If $a_n > S'$, it means that

$b_n = 0$ and proceed to a_{n-1}. Otherwise $b_n = 1$, then set $S' = S' - a_n$ and compare with a_{n-1} to determine b_{n-1}. Repeat the procedure and the sequence (message block) $\{b\}$ can be found. It is found that the original knapsack algorithm was broken and many variant of knapsack algorithms have been proposed but some of them have also been broken [34].

8.3.3 RSA

The RSA cryptosystem is one of the most widely used public key cryptosystem nowadays. The system was first introduced by R. L. Rivest, A. Shamir and L. M. Adleman in 1978[35]. To generate the encryption and decryption keys, first generate two large random primes p and q and compute $n = pq$. Then generate a random integer e such that $gcd(e, \phi(n)) = 1$ $(\phi(n) = (p-1) \times (q-1))$. The numbers n and e are the encryption keys and made public. The decryption key d is defined as the inverse of $e \pmod{\phi(n)}$, i.e. $e \times d \equiv 1 \pmod{\phi(n)}$. The decryption key d and the primes p, q are kept secret.

To encrypt a message, first breaks the message into many integer blocks M_i such that the value of each block is between 0 and $n-1$. The ciphertext C_i is,

$$C_i = E(M_i) = M_i^e \pmod{n}$$

To decrypt the ciphertext,

$$M_i = D(C_i) = C_i^d \pmod{n}$$

The theory behind RSA is based on Fermat's Little theorem, if p is a prime and does not divide M, then $M^{p-1} = 1 \pmod{p}$. Since $e \times d \equiv 1 \pmod{\phi(n)}$, therefore $e \times d = k\phi(n) + 1$ for some integer k. From Fermat's Little theorem,

$$\begin{aligned} M^{k\phi(n)+1} &\equiv M^{[k(q-1)]\times(p-1)+1} \equiv M \pmod p \\ &\equiv M^{[k(p-1)]\times(q-1)+1} \equiv M \pmod q \end{aligned}$$

From these two equations, it implies that for some integers r, s,

$$M^{k\phi(n)+1} = M + rp = M + sq.$$

Since p, q are primes, so $r = tq$ and $s = tp$ for some integer t. Therefore,

$$M^{k\phi(n)+1} = M + tpq = M \pmod n,$$

and for all $M_i, 0 \leq M_i < n$,

$$M_i^{e\times d} \pmod n \equiv M_i^{k\phi(n)+1} \pmod n \equiv M_i \pmod n$$

8.3.4 Elliptic Curve Cryptosystem

An elliptic curve over $\mathcal{Z}_n$ and $gcd(n, 6) = 1$ is defined by,

$$E : y^2 = x^3 + ax + b,$$

where a and b are integers and $gcd(4a^3 + 27b^2, n) = 1$. The points of E includes the solutions $(x, y) \in \mathcal{Z}_n \times \mathcal{Z}_n$ and a point $\mathcal{O}$ called the point at infinity.

For points $P = (x_1, y_1) \in E$ and $Q = (x_2, y_2) \in E$. If $Q = -P = (x_1, -y_1)$, then $P + Q = \mathcal{O}$. Moreover, $\mathcal{O} + P = P + \mathcal{O} = P$, which means that $\mathcal{O}$ is the *identity element.* If $Q \neq -P$, then $P + Q = (x_3, y_3)$ with

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 \\ y_3 &= \lambda(x_1 - x_3) - y_1, \end{aligned} \qquad \text{and} \qquad \lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{if } P \neq Q, \\ \dfrac{3x_1^2 + a}{2y_1}, & \text{if } P = Q. \end{cases}$$

The points on E over a finite field K form an abelian group and the addition operations involve are simple and easy to implement. The elliptic curve logarithm problem is believed to be harder that the discrete logarithm problem in finite field of the same size of K. In 1985, N. Koblitz and V. Miller proposed the use of elliptic curve to public key cryptosystem. Elliptic curves over finite fields can be used to implement existing algorithms like the **Elgamal**, **Schnorr** and **Digital Signature Standard** [36].

8.3.5 Public Key vs. Private Key Cryptosystem

As described above, the two main advantages of public key cryptosystem is that they provide better key management and distribution, and a new way of authentication - digital signature. However, one main drawback of public key is that they are slower than private key cryptosystems because of the complex computation involved like modular arithmetic and prime number generation. In fact, we can use public key cryptosystems for key exchange and distribution, and authentication. Meanwhile, use private key cryptosystems for data encryption and decryption to obtain the best performance. Therefore, it maybe difficult and unfair to compare them based on the security, since it greatly depends on the key length [18, 32].

8.4 Digital Signature

Signature is a very important means for validation in business world. In order to use **digital signature** in electronic document like written signature for paper document, we need algorithms to verify the authenticity of a message. The digital signature should be able to be verified by anyone and cannot be forged. Moreover, the signer cannot deny that his has signed the message. Nevertheless, digital signature is different from written signature that it varies for different document, otherwise, the signature can be attached to any message. In private key cryptosystem, authentication is not easy to achieved, this problem can be solved by using public key cryptosystem [35].

Suppose user A wants to sign a message M to user B, A first uses his private key to sign (decrypt) the message to $S = D_A(M)$ and sends it to B. When B received the signature S, he looks up A's public key and verifies (encrypts) S by this encryption key, $E_A(S) = E_A(D_A(M)) = M$. In this way, A cannnot deny the signature since only he knows the private key. Also, B can verify that A signed the message M and he cannot modify M since it is difficult for him to create another message $M' \neq M$ such that $D_A(M') = S$.

However, this simple protocol may have some problems : (i) The file containing A's public key may be tampered and so the verification of A's signature with the wrong public key will become meaningless. (ii) If A's private key is compromised and made public (either accidentally or intentionally), then anyone can pretend A to sign message using A's key and A has to respond for these signed message. (iii) B is able to copy this signed document and reuse it [33].

The first problem can be solved by authenticating the public key by a trusted third party called the **Central Authority** (CA). The CA issues *certificates* C_A for A containing A's identity and public key all signed by the CA's private key. B can obtain A's certificate and compute the contents using the public key of the CA. The second and the third problem can be solved by using a *timestamp.* A timekeeper will timestamp the message M by signing M and the current time with timekeeper's private key. B can ask the CA for the validity of A's public key based on the timestamp. If A lost his private key, he can report the CA for the lost so only message with timestamp before A's report are still valid.

Some popular digital signature scheme based on public key cryptosystem include the **ElGamal Signature** scheme and the **Digital Signature Standard.**

8.4.1 ElGamal Signature Scheme

The **ElGamal public key and signature scheme** was proposed by T. ElGamal in 1985 based on discrete exponentiation. Using this signature scheme, to generate the signature for a message M, first select a large prime p. Then generate a random integer k such that $gcd(k,p) = 1$ and choose an element $\alpha \in \mathcal{Z}_p^*$. Compute $m = H(M)$ (H is an one-way hashing function) and $r = \alpha^k$ (mod p). Finally, solve for s in equation,

$$\begin{aligned} m &= ar + ks \pmod{p-1} \\ s &= (m - ar)k^{-1} \pmod{p-1}. \end{aligned}$$

The pair (r, s) will be the signature for message M. The private key will be a and the public keys are α, α^a and the prime p.

To verify the signature (r, s) of the message M, first compute,

$$r^s \alpha^{ar} = \alpha^{ks} \alpha^{ar} = \alpha^{ar+ks} \pmod{p}.$$

Then compute and validate if $\alpha^m = r^s \alpha^{ar} \pmod{p}$ [18, 36].

8.4.2 Digital Signature Standard (DSS)

The **Digital Signature Standard** (DSS) is first proposed by the U.S. National Institute of Standards and Technology (NIST) in August, 1991. Using DSS, to sign a message M, first select a large prime p where $2^{511} < p < 2^{512}$ and a prime divisor q of $p - 1$ with $2^{159} < q < 2^{160}$. Then generate a random integer k with $0 < k < q$ and choose an element $g \in \mathcal{Z}_p^*$ of order q. Select the private key x with $0 < x < q$ and compute the public key $y = g^x \pmod{p}$. Compute $m = H(M)$ ($H : \mathcal{M} \rightarrow \mathcal{Z}$ is an one-way hashing function) and $r = (g^k \pmod{p}) \pmod{q}$. Finally, solve for s in equation,

$$\begin{aligned} m &= -xr + ks \pmod{q} \\ s &= (m + xr)k^{-1} \pmod{q}. \end{aligned}$$

The pair (r, s) will be the signature for message M and the public keys are p, q, g and y.

To verify the signature (r, s) of the message M, first compute,

$$w = s^{-1} \pmod{q}, \qquad u_1 = mw \pmod{q} \qquad \text{and} \qquad u_2 = rw \pmod{q}.$$

Then compute and validate if $r = (g^{u_1} y^{u_2} \pmod{p}) \pmod{q}$. The process of the verification is that,

$$k = mw + xrw \pmod{q} = u_1 + xu_2 \pmod{q}.$$

Therefore,

$$\begin{aligned}(g^{u_1}y^{u_2} \pmod p) \pmod q &= (g^{u_1}g^{xu_2} \pmod p) \pmod q \\ &= (g^k \pmod p) \pmod q = r\end{aligned}$$

Just like Diffie-Hellman key exchange, the security of the ElGamal and DSS signature scheme are also based on the difficulty of discrete logarithm. Comparing with ElGamal signature for the same value of p, the size of DSS signature which depends on q (order of a subgroup of $\mathcal{Z}_p^*$) is smaller [18, 36].

8.5 Cryptanalysis to Current Cryptosystems

8.5.1 Differential Cryptanalysis

A well-known cryptanalysis to DES and some DES-like cryptosystems is **differential cryptanalysis** developed by E. Biham and A. Shamir [37, 38]. It is a chosen-plaintext attack that uses 2^{36} ciphertexts to compute the key instead of 2^{55} complexity brute force attack for 16 round DES. These ciphertexts are selected from 2^{47} chosen plaintexts by a simple bit repetition criteria. This scheme can also work even the selected ciphertexts are computed from different keys.

In this attack, the initial permutation IP and final permutation IP^{-1} of DES are ignored since they have no effect on cryptanalysis. The basic idea of differential cryptanalysis is to analyse the relation between the differences (*XOR* for DES-like cryptosystems) in plaintext pairs and the differences of their corresponding ciphertext pairs. The differences will then be used to estimate the most probable key.

In DES, when a particular XOR value of two plaintexts are modified simultaneously, the particular XOR value of intermediate round will be modified in certain way with relatively high probability (two such encryptions are called a pair). The XOR of pairs is linear in operations of EP, P and $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$. Moreover, the nonlinear S-box in DES provides a probabilistic distribution of possible output XORs for each input XOR of which many output XORs have a relatively high probability. Based on this property and a pair of chosen ciphertexts, a set of possible subkeys K_i for the function f in the last round can be found. Moreover, the input and output XOR of the function f and each S-box in the last round can also be found. With only few pairs, most subkey values can be estimated [37].

Differential cryptanalysis can also be used to other DES-like cryptosystems. For example, the Lucifier with eight rounds can be broken within 60 ciphertexts and the Feal-8 can be broken within 1000 pairs of ciphertexts.

8.5.2 An Attack to RC4 in Netscape1.1

The idea of the attack to RC4 in Netscape1.1 is based on the insecure of random number generator used by Netscape1.1. The session key can be determined in 25 seconds and this attack also work for 128-bit version [H]. This attack was proposed by two graduate students at the University of California at Berkeley, and published on the Internet in Sep 1995. [I]. Their idea is that Netscape uses a combination of the time seconds and mircoseconds, the process ID and the parent process ID to seed the random number generator. They proposed that approximated values can be determined, for example, from time network daemon and mail daemon. Appropriate guessing of these values may reduce the expected search to less than brute-force.

According to Netscape's response, this vulnerability exists because the size of random input is less than the size of the subsequent keys. Hence a brute force search of a smaller key space is needed instead of all the 2^{128} possible keys. They claimed that the problem have already been fixed and they planned to increase the random information from about 30 bits to about 300 bits. Moreover, they said that this discovery does not affect the security of SSL, RC4 or other Netscape's security implementations [J].

However, Community ConneXion criticized Netscape's response that it is inappropriate for Netscape to say that they use random information to generate session encryption keys since these information increases sequentially at a known or predictable rates. Furthermore, they said that the vulnerability exists is not due to the size of random input. In fact, the security flaw because the initial input to the algorithm is not random. The fact that the size of the non-random input is less than the subsequent keys is another factor that lowered Netscape's security [H].

Though Netscape has increased randomness in their new version of browser, from this attack, we can know how important the randomness of a key is.

8.5.3 An Timing Attack to Diffie-Hellman, RSA

According to [K], cryptanalysts can use time measurement of cryptographic operations to reveal key materials since it is hard to software implement an algorithm that run in fixed time. It means that for different inputs, the process time may also different. They can use known ciphertext attack with timing measurements to find out the secret exponent of Diffie-Hellman system, factor RSA keys n and so on.

1. Fixed exponent Diffie-Hellman - Timing measurement can be used to find the secret exponent x that uses for multiple Diffie-Hellman key exchange. Suppose the opponent has the knowledge of values n, α and the general design of the target system. Then he can measure the time to compute $X = \alpha^x \pmod n$ over several operations. In the modular multiplication, if a bit x_i is 1, the intermediate step $y = y \times \alpha \pmod n$ is computed. Otherwise, this step will be skipped. This time difference can be used to estimate the bit x_i.

2. Factoring RSA private key - For the RSA operation $M = C^d \pmod n$ not using Chinese Remainder Theorem, the same attack as for Diffie-Hellman can be applied here. In Chinese Remainder Theorem, $C \pmod p$ and $C \pmod q$ are first computed which do not run in constant time. The known ciphertext attack is to choose values of C close to p (or q) and measure the time to compute $C \pmod p$. If $C < p$, there is not effect to compute $C \pmod p$, if C is slightly greater than p, $C \pmod p = C - p$.

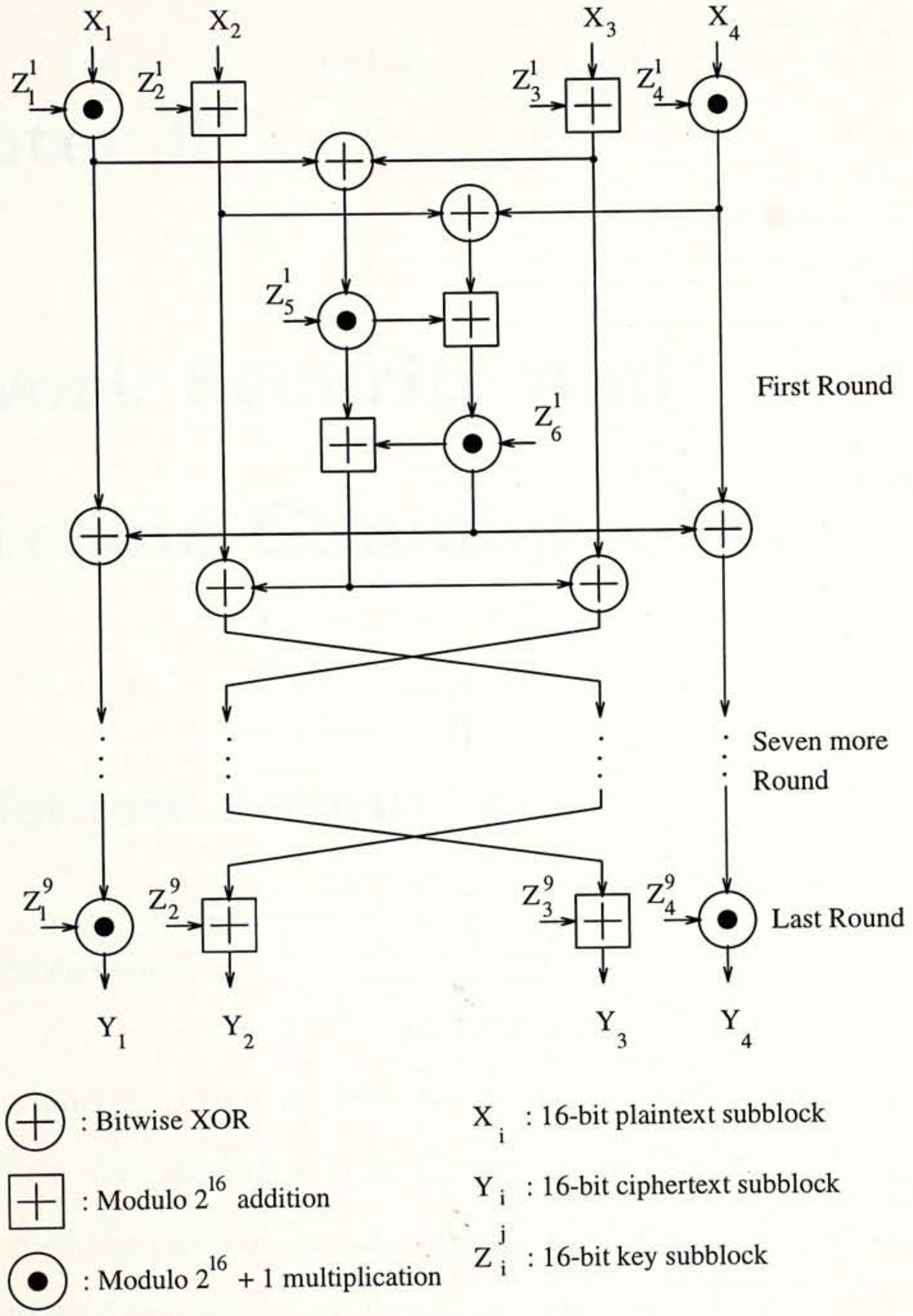


Figure 8.2: The Encryption and Decryption algorithm of IDEA

Chapter 9

Network Security and Electronic Commerce

9.1 Network Security

9.1.1 Password

One very important thing in network security is **password**. A password is used to inform the network system about the identity of the user. Therefore, the selection of the password greatly affects the security of users and networks. Some selection of password that should prevent [18, 39]:

1. Use the username, account name, phone number, birthday or similar information of relevant person.

2. Use words from different databases. For example, use words in English or foreign dictionary, names of place, proper noun, string of same letter, special keyboard patterns like "qwert" and so on.
3. Use letter permutation, capitalization or word pluralization to the words obtained from the first two steps.
4. Use the same password over many different accounts.

Meanwhile, here are some suggestions to select and use passwords [39]:

1. Use digits, punctuation characters with uppercase and lowercase letters.
2. Use password which is easy to remember but difficult to guess. For example, combined two or three short words separated by digit and punctuation character, use some special acronym of a long phrase.
3. Change the password frequently like once every few months. Hence even the password is learnt by someone, it will change after a certain time.

SecurID

The **SecurID** is a user identification and authentication system developed by the Security Dynamics Technologies, Inc. Security Dynamics was founded in 1984, and is going to acquire RSA for their security products. This system uses two levels of authentication, (i) the tradition reusable password or the personal identification number (PIN) and (ii) an access control security token called the SecurID token.

With Security Dynamics' access control modules (ACMs), the SecurID token generates an one-time-only, unique and unpredictable access code every 60 seconds. This code will be displayed on the SecurID token so that only those who has the token can gain access. The SecurID token will synchronize with a hardware or software ACM. To access a computer, a user first enters his secret PIN and then the displayed code on the SecurID token. Authentication is done if both the unique PIN and the token's code are matched [L].

Smart Card

A **smart card** is a tamper-resistant plastic card that stores and protects secure information on an embedded intergrated microprocessor chip. There are two main types of smart cards, (i) an "intelligent" smart card with CPU, it has read, write and decision making ability, so new information can be added and processed and (ii) a memory card which mainly for information storage like the stored value a user can spend.

The intelligence of the IC chip can protect the stored information from attack. Hence smart cards are more secure than tradition magnetic stripe cards that store information outside the card and easier to be copied. Moreover, a smart card can use a secret key such as PIN to match the access code like that for ATM card. Smart cards can be used for secure access to networks, storage of secret keys, digital signatures and electronic cash [M].

9.1.2 Network Firewalls

One common technology today for Internet security is the use of **firewalls**. A firewall can be considered as a intermediate system between the Internet and a trusted network. It acts as a filter that only permits authorized information to throw into or out of the network [22]. A firewall can be implemented by setting up routers, computers or hosts to protect the local network from unwanted connections or information outside the network [40, N].

Without firewall, the security of the network may only rely on the host security. A firewall can filter out some insecure services like NFS or NIS and attack from hosts outside the protected network. Besides, a firewall can enforce a network access policy by controlling the access to the network such that a host can be made unreachable from unwanted access outside the network. Moreover, a firewall can protect some Domain Name Server (DNS) information like IP addresses and names. Also, it can provide log and statistics about network access and usage to monitor the network.

Though firewall has several advantages, there are still some potential problems. Firstly, firewall cannot provide security within the network. Besides, it cannot prevent the transfer of virus-infected files into or out of the network. Moreover, as the security of the network is concentrated in the firewall, any problems or flaws in the firewall will substantially affect the entire network. Since all connections must be filtered by the firewall, it can also be a bottleneck of the network and limits the throughput [N].

The firewall system use two levels of network policy (the network access policy and the service access policy) to define the services allowed and denied for the network. The main components to implement these policies include :

Packet-filtering routers

A **packet-filtering router** filters IP packets during input and/or output time based on their source or destination *IP addresses*, or in some routers, the source or destination *TCP/UDP ports.* A packet-filtering router can block the connections to or from certain hosts, networks or ports. Moreover, it can control the kind of services that are accessible by other hosts. For example, a site may allow any hosts to use SMTP but allow only one Telnet connection to a host. Some services which are inherently dangerous are usually blocked by the firewall to or from any hosts. Besides, for those services which are less risky, the firewalls will restrict the access to only certain systems.

There are some problems using packet-filtering routers, firstly, the ruleset is complex especially when exceptions to rules are required for certain services. If the routers do not filter on the TCP/UDP source port, the ruleset may become even more complex. Secondly, some services use random port numbers, they include *tcpmux protocol* and *portmapper* used by RPC. The router cannot block these services unless it blocks all possible ports, however, it would also block some desired services [40].

Application-level gateways

An **application gateway** is a host that uses an application called a *proxy service* to forward and filter connections for services like Telnet and FTP. Using proxy services, only those services with a proxy can go through the application gateway. For example, if the application gateway contains proxies for FTP and SMTP, then all services except FTP and SMTP are blocked. Besides, using proxy services, the protocol can be filtered.

Moreover, security can increase by logging and authenticate the connection. Application gateways can be used together with other gateways like packet filtering routers and circuit-level gateways to increase flexibility and security. The filtering rules will become less complex than the packet filtering router [40].

Circuit-level gateways

It is sometimes considered as a kind of application gateway. **Circuit gateways** relay TCP connections, the source host connects to a TCP port on the gateway and connects to the destination on the other side. Using circuit gateways, once the connection between the source and the destination is setup, the gateway will just act as a wire that passes the information between the two hosts. The gateways can monitor and control the connection by logging (like number of bytes flow and TCP destination), limit the duration of the connection, maintain a list of allowable external hosts and so on [40].

9.2 Implementation for Network Security

9.2.1 Kerberos

Kerberos is a trusted third-party authentication protocol based on Needham and Schroeder's protocol and later modified by Denning and Sacco. The original work is done by Project Athena and the Massachusetts Institute Technology (MIT). Kerberos provides verification of the identities on an open network without relying on the authentication of host operating system, addresses and physical security [O].

The authentication process includes:

1. A client sends request to the *authentication server* (AS) for a **credential** of a given server. The credential includes a ticket of the server and a session key. There are two approaches, using the first method, a client sends a request for a *ticktet-granting ticket* (TGT) for later use with the *ticket-granting server* (TGS). The second method is that the client sends the request to the TGS.
2. For the first approach, the AS responds by the client's key (derived from the password) encrypted credentials. For the second approach, the responds is encrypted by the session key from the TGT. The ticket has the client's identity, session key, a timestamp and other information all encrypted by the server's secret key.
3. The client sends the ticket to the server to verify the identities. An addition

information called *authenticator* is used to prove the message is sent by the client/server to whom the ticket was issued. This authenticator is encrypted by session key with a timestamp.

4. The session key is used to authenticate the client and optional the server and can also be used to ensure the integrity and privacy of the message exchanged. The session key is temporary used within a single login session.

The encryption system is Data Encryption Standard (DES) in Cipher Block Chaining (CBC) mode with a Cyclic Redundancy Check (CRC-32), MD4 or MD5 checksum to the message sequence. In CBC mode, initialization vector (IV) of zero is used.

9.2.2 Privacy-Enhanced Mail (PEM)

Privacy-Enhanced Mail (PEM) is developed by the Privacy and Security Research Group (PSRG) of the Internet Resources Task Force (IRTF) and the PEM Working Group of the Internet Engineering Task Force (IETF). PEM provides privacy enhancement services like authentication and message integrity for electronic mail transfer over the Internet [P].

PEM uses two-level keying hierarchy for encryption:

1. **Data Encrypting Key** (DEK) - for encryption of message text and computation of **Message Integrity Check** (MIC). When using public key, it is also used for encryption of signed representation of MIC in a PEM message.

2. **Interchange Key** (IK) - for encryption of DEKs. When using private key, the IK is the private key shared between the sender and the receiver. When using public key, the IK component for DEK encryption is the public key of the receiver whereas the component for MIC encryption is the private key of the sender.

The plaintext message is first accepted in local form and then converted to a *canonical message* defined as equivalent to the SMTP representation. This canonical form will be used for MIC computation and encryption if required. The output is then combined with a header containing the cryptographic control information. The output PEM message is sent as the text portion to the electronic mail system for transmission. When the message is received, the information in the header is used for MIC validation and decryption. The canonical output is converted to local form.

Moreover, the algorithms and modes used by PEM mainly include [Q]:

1. *Message Encryption* algorithms - DES with cipher block chaining (DES-CBC) for encryption of message text and signature (using public key).
2. *Message Integrity Check* algorithms - use MD2 and MD5 message digest.
3. *Symmetric Key Management* algorithms - DES in Electronic Codebook (DES-ECB) mode and DES in Encrypt-Decrypt-Encrypt (DES-EDE) mode with pairs of key for encryption of DEK and MIC.
4. *Asymmetric Key Management* algorithms - RSA is used for encryption of DEK and MIC, and RSA with MD2 is used to sign certificates and

certificate revocation list (CRL).

9.2.3 Pretty Good Privacy (PGP)

Pretty Good(tm) Privacy (PGP) is developed by P. Zimmermann from Phil's Pretty Good Software. PGP uses public key encryption for people to exchange messages with privacy, authentication and convenience [41]. PGP uses RSA public key with IDEA private key encryption, MD5 message digest for digital signatures and data compression before encryption.

To protect the secret even it is known by someone else, PGP asks for a *pass phrase* before one can access the private key file. The pass phrase can be a whole phrase or sentence which is like a password. The generation of the public and private key is based on random numbers derived mainly from measuring intervals between keystrokes using a fast timer. Public keys are kept in key certificates with user ID, key generation timestamp and the key. The public and *private key rings* contains one or more of these key certificates. In this way, instead of keeping keys in separate key files, the use of key rings facilitates the automatic lookup of keys by user ID or key ID (an abbreviation of the public key for internally reference of the key).

Encrypted files are prefixed by the key ID of the public encryption key and the receiver can use it to look up for the private decryption key. Signed document are prefixed by the signature certificates containing the key ID of the signed key, signed message digest of the document and the timestamp of signature creation time. The receiver can use the key ID to look up the sender's public key for signature verification.

The confidence of ones public key can be improved by keeping the public key with a collection of *introducers'* certifying signatures. One can become an introducer by signing someone public key and return it with his signature. PGP keeps tracks of the keys on the public key ring that are certified with signatures from a trusted introducers. When PGP validating a public key, it examines the trust level of the certifying signatures and computes the weighted score of validity, for example two marginally trust signatures may be considered as a trusted signature.

9.3 Internet Commerce

The advantages of Internet commerce includes [42]:

1. Information update in Internet is more efficient and less expensive since the merchants do not need to reprint and redistribute the information.
2. Less printing and distribution cost of sending information or advertisement through Internet than tradition method by printed catalog or mail.
3. Environmental protection because of less consumption of paper.
4. The merchant can promote their products no only by means of printed matter, but also other media like audio or video.
5. Interactive service between the consumers and merchants. The consumers can send feedback or purchase order through on-line Internet service.

Meanwhile, to provide a better environment for Internet commerce, here are some desired requirements for Internet commerce system [AP]:

1. *Security* and *Reliability* - since Internet is an open environment, a payment system which involves transfer of money and private information must be secure and reliable. In electronic cash system, illegal creation, copying or double spending of electronic cash should be prevented or detected.

2. *Scalability* - as the growth of Internet is rapid, the payment system should be able to handle the increment of consumers and merchants without significant performance degradation.

3. *Anonymity* - in some transactions, the identity of the parties participate should be untraceable.

4. *Acceptability* - the payment systems should be supported by multiple servers and banks so that transactions between users of different servers or banks become possible.

5. *Flexibility* - the payment systems is desired to support different payment methods analogous to credit cards, cheques and paper cash.

6. *Transferablity* - a user can spend electronic cash with a third party without first connecting to the cash server.

7. *Interoperability* and *Convertibility* - since there exists several methods of payment in the Internet, the funds in one system should be convertible into funds in other systems.

9.3.1 Electronic Cash

Some common payment methods for Internet commerce includes *credit card payment* and *electronic cheque.* However, these services are non-anonymous and the identity of the parties of the transaction are not protected. **Electronic cash** is a new concept of payment, the users can use electronic cash for transaction like paper cash in which their identity are anonymous and untraceable. This concept is based on **blind signatures** proposed by D. Chaum. A user can blind a document and ask for a digital signature, the signer will have no idea about the identity of the sender.

Blind Signature

The concept of **blind signature** is first proposed by D. Chaum in [43]. Suppose a user A requests another user B to sign a document x without knowing the content of the document. Use blind signature protocol A first generates a random value r called the *blind factor*, and uses r to compute a *commuting function* $c(x)$ and sends to B. Then B signs $c(x)$ by the private key and return $D_s(c(x))$ to A. A retrieves $D_s(x)$ by applying $c^{-1}(D_s(c(x))) = D_s(x)$ and verifies the signature by $B's$ public key and checks if $E_s(D_s(x)) = x$. The commuting function and its inverse are only known to A and the signing function deploys digital signature scheme based on public key cryptography.

Since the signer B does not know the content of the document he signs, there is a risk that A may cheat B by sending a document which is adverse to B. There is a technique to prevent A from cheating B. First A make n copies of the document and each is blinded with a different blinding factor. A sends all these documents to B. B randomly chooses $n-1$ of them and asks A for the blinding factors for these documents. B extracts the documents using the received blinding factors and checks if all these $n-1$ documents are correct. If yes, B signs the document remained and returns to A. In this way, the probability that A can cheat B is much smaller and can be reduced by a larger n [18].

The blind signature protocol can be applied to construct a untraceable payment system. Suppose a user A requests the bank to sign a electronic note x with anonymity and uses the note for electronic payment, the procedures are [AD]:

1. A generates a random blinding factor r and computes $c(x) = xr^e \pmod{pq}$ and sends to the bank. The parameters e is the bank's private key, p and q are two large prime numbers, they are generated and used in the same way as in RSA public key cryptosystem.

2. The bank signs the note with $(xr^e)^d = rx^d \pmod{pq}$, return the signed note to A and debits $A's$ account. The bank cannot determine x since the value of the blinding factor r is not known.

3. A retrieves the signed note by $c^{-1}(rx^d) = (rx^d)/r = x^d \pmod{pq}$. Then A verifies the note by the bank's public key e and checks if $(x^d)^e = x \pmod{pq}$ and stores $x^d \pmod{pq}$ if valid.

4. When A decides to pay, he sends $x^d \pmod{pq}$ to the merchant.

5. The merchant verifies the note and forwards to the bank if valid.

6. The bank verifies the note and checks if the note is already on the list of cleared notes. If the checking are correct, the bank credits the merchant's account and informs the merchant.

9.4 Internet Browsers

9.4.1 Secure NCSA Mosaic

NCSA Mosaic is one of the earliest Internet information browser and World Wide Web (WWW) client. NCSA Mosaic was developed at the National Center for Supercomputing Applications (NCSA) at the University of Illinois in Urbana-Champaign. **Secure NCSA Mosaic** with security enchanced version developed by NCSA with Enterprise Integration Technologies (EIT) and RSA Data Security Inc. (RSADSI) [R]. Through the implementation of **Secure HyperText Transfer Protocol** (S-HTTP), authentication, privacy, certificate and key management are provided.

Secure NCSA Mosaic uses RSA for key exchange between client and server and generates RSA key pairs from 512 to 1024 bits and public key certificate. A user can also obtain public key certificates from Public Certificate Issuing Agency and he can digitally sign and verify messages to and from the server.

Secure HyperText Transfer Protocol (S-HTTP)

Users can securely communicate with servers using Secure HTTP Uniform Resource Locator (URL) **shttp** instead of **http**. So the users should use "shttp://" to connect to a secure servers or just use "http://" for non-secure WWW like NCSA Mosaic. S-HTTP supports end-to-end secure transaction and allows the server and the client to negotiate a wide range of operation modes, encapsulation formats, key management, encryption and signature algorithms.

In S-HTTP, message is protected by : (i) signature (ii) encryption - with two key exchange algorithms, by *public key inband key exchange* or externally arranged keys and (iii) authentication - verify message integrity and sender authenticity through the computation of a *Message Authentication Code*, a keyed hash over the document [S]. The negotiation parameters for S-HTTP are:

1. *Privacy Domain types* - PKCS-7, PEM and Pretty Good Privacy (PGP).
2. *Certificate types* - the types of Public Key Certificate that the agent will accept which include X.509 and PKCS-6.
3. *Key Exchange algorithms* - RSA, Outband (certain external key agreement), Inband (direct assignment of an uncovered key to a symbolic name) and Kerberos.
4. *Digital Signature algorithms* - RSA and NIST-DSS.
5. *Message Digest algorithms* - MD2, MD5 and NIST-SHS.
6. *Private Key algorithms* for message - DES, IDEA, RC2, RC4 and CDMF.

7. *Private Key algorithms* for header - DES, IDEA, RC2 and CDMF.

8. *Privacy Enchancements* - indicate whether a message is signed, encrypted or authenticated.

9.4.2 Netscape Navigator

Netscape Navigator is one of most popular commercial network navigator developed by Netscape Communications. They also developed another product called Netscape Commerce Server to provide secure electronic commerce and communications services over Internet and other TCP/IP-based network [T].

Netscape Communications has designed a security protocol called **Secure Sockets Layer** (SSL) to provide data security layered between TCP/IP and application protocols like HTTP, Telnet, NNTP, or FTP [U, V]. Users can connect to secure HTTP servers using SSL by a new URL access method **https** instead of **http**. So "https://" should be used when connecting to HTTP URLs with SSL or just use "http://" as before for HTTP URLs without SSL. The cryptographic technologies of Netscape Navigator are mainly developed by RSA Data Security Inc. They use 128-bit key size RC4 stream encryption algorithm within U.S. and a 40-bit key size version is approved for export to most countries.

Secure Socket Layer (SSL)

Secure Sockets Layer (SSL) Protocol is a two-layer Internet security protocol [W]. The lower layer (*SSL Record Protocol*) layered above a reliable transport protocol like TCP/IP for encapsulation of various higher level protocols. Before data transmission, user can first use *SSL Handshake Protocol* to negotiate the desired encryption algorithm and session key and SSL can also provide server and client authentication. Thereafter, any data exchange between application protocols will be encrypted and decrypted by SSL. Different application protocols can then be layered on top of SSL Protocol transparently.

The SSL protocol can provides channel security in three aspects [T, X],

1. After the private key is negotiated by initial handshake, all data will be encrypted by private key cryptography (like DES, RC4).
2. The server and client identity are authenticated. A client can verify the identity of a server using certificate and digital signature. The authentication process is done by public key cryptography.
3. The data transport includes a data integrity check using a MAC by secure hash functions (like SHA, MD5).

The main process of SSL Handshake protocol is,

1. The client sends a *client hello* message to the server and the server should respond with a *server hello* message. This message exchange defines the client/server version, session ID, cipher suite and compression method.
2. The server will send its certificate if it is to be authenticated. The *server key exchange* message may be sent if required.

3. The client must send either *certificate message* or a *no certificate* alert if a *certificate request* is received. Based on the public key algorithm selected during the exchange of hello messages, the *client key exchange* message (RSA, Diffie-Hellman or Fortezza-KEA) is sent.

4. A *certificate verify* message may be sent to verify the certificate.

5. The client sends a *change cipher spec* message and the *finished* message based on the new algorithm, keys and secrets.

6. The server responds by sending its *change cipher spec* message and then sends its *finished* message. The handshake is now completed and application layer data can start exchange between the client and the server.

9.4.3 SunSoft HotJava

The **HotJava** browser is a new Internet navigator developed by Sun Microsystems, the main feature of HotJava is the interactive services. The users can access application softwares dynamically without installing them, which means that the softwares are transparently migrated over the network. The uses of HotJava's dynamic services includes [Y]:

1. *Interactive Content.*

2. *Dynamic Types* - can understand different types of objects through dynamically link the Java code from the host to support the object.

3. *Dynamic Protocols* - uses protocol name to link appropriate handlers instead of built-in protocol handlers, so new protocols can be incorporated dynamically.

The reason that HotJava can provide these dynamic capabilities is that HotJava is written in a new language called **Java(tm)**. Java is a dynamic and object-oriented language which is close to C++. Moreover, as Java is used in distributed environment like Internet, security is important. Java can be used to construct a virus-free and tamper-free system and it uses public-key encryption for authentication [Z].

Security of HotJava

Using HotJava, code fragments are capable to import, install and execute over the Internet transparently. Hence, security problem is significant to HotJava and it has four layers of interlocking facilities to provide secruity [AA]:

1. The Java language and compiler - Pointer arithmetic and modification are not allowed in Java. Java uses true arrays instead of pointer arithmetic.
2. Verification of bytecode - Verification is needed to clarify if the import code fragments come from a trustworthy compiler. The import code cannot executed before it has passed **verifier's** test and after the verification :
    (a) There are no operand stack overflows or underflows.
    (b) The types of the parameters to all opcodes are correct.
    (c) There are no illegal data conversions.

(d) All object field accesses are legal.

Therefore, using HotJava, a private variable can be trusted as really private. Moreover, file access and the ability of applets to grab files and throw over firewalls are also restricted.

3. The **Class loader** - The thread of execution running Java bytecode can be considered as a set of classes partitiion into *namespaces*. The class loader guarantees that unique namespace exists for built-in class (class from the local file system) and each network source. To prevent an imported class to affect a built-in class, the runtime system first checks the class in the built-in's namespace and then the referencing class's namespace.

4. Interface-specific security - The security of the first three lower levels enable all local classes protected from modification by imported code. To network level, HotJava uses mechanisms to provide information about the trustworthiness of imported code. For example, it can check whether the origin of the code is inside or outside a firewall or in future uses public key cryptosystem with message digest to check the origin and integrity of the code.

Chapter 10

Examples of Electronic Commerce System

Current electronic payment methods in Internet include [AP] :

1. *Electronic Cash* - has main advantage of its anonymity but the main problem is the requirement of a large database to track past transactions so as to prevent double spending. Current electronic cash-based systems include **DigiCash** and **USC-ISI's NetCash**.

2. *Credit-debit model* - it includes **First Virtual**, **NetBill** and **USC-ISI's NetCheque**. Consumers are registered with accounts on commerce server and debit the account for the payment. This model has an advantage of auditability, therefore, it is not typically anonymous.

3. *Secure Credit Card* - the model in which the consumers only need the credit card numbers and do not need to be registered with the payment system. The consumers' credit card numbers and purchased amount are encrypted by public key such that only the merchants or the payment processing service can read them.

10.1 CyberCash

CyberCash, Inc. was founded in August 1994 by B. Melton, D. Lynch, S. Crocker, M. Yesil and B. Wilson to provide a secure and convenient electronic payment system over the Internet. With the CyberCash Secure Internet Payment System, secure transactions on the Internet between consumers, merchants and their banks are done. CyberCash claimed that they are the only company with world-wide export license of 768-bit RSA encryption algorithm. They also use 56-bit DES for encryption and all transactions are authenticated using MD5 with 768-bit RSA signatures. CyberCash transactions are transferred over three separate softwares located at the consumer's PC (called a *wallet*), the merchant's server and the CyberCash servers [AB].

Credit Card and Money Payment

CyberCash provides both secure credit card and money payments services on the Internet. The main procedure of **Secure Internet Credit Card Payment** is shown in Figure 10.1 [AC]:

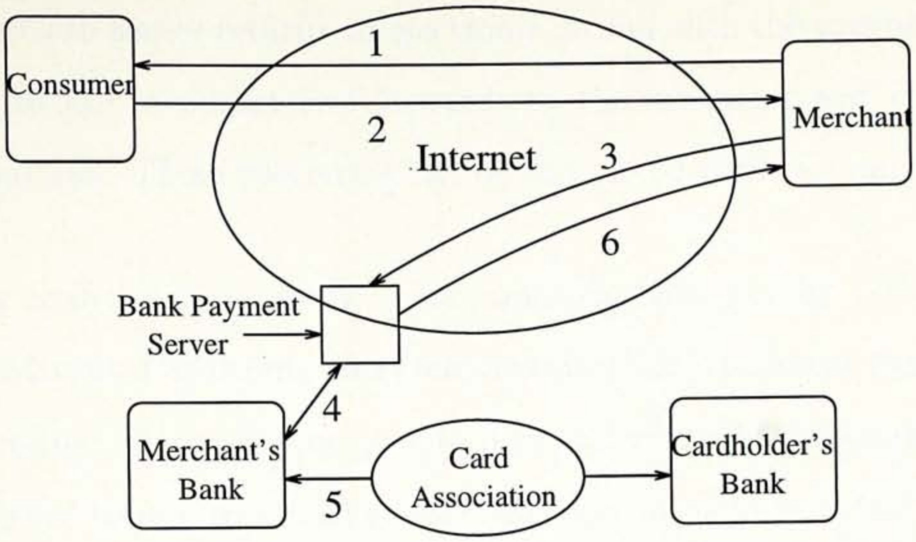


Figure 10.1: CyberCash Secure Internet Credit Card Payment

1. The consumer goes to a merchant's site and selects the goods or services he wants to purchase. Then the merchant server returns a summary of the items like the price and transaction ID to the consumer.

2. When the consumer is ready to pay, he chooses a "wallet" and which credit card from their "wallet" to pay. Then he can forward the order and encrypted payment information to the merchant.

3. The merchant extracts the order part, digitally signed and encrypt the encrypted payment information by his private key and forwards to the CyberCash server.

4. CyberCash server examines the received information and forwards to the merchant's bank over dedicated lines.

5. The merchant's bank forwards the credit card authorization data to the cardholder's bank through the card association and returns the approval or denial code back to the CyberCash server.

6. CyberCash server returns an electronic receipt with the approval or denial code to the merchant and forwards to the consumer and complete the transaction. These procedure can be completed with one minute.

Besides credit card payment, a consumer can also pay by cash using **CyberCash Money Payments Service**, he can make a payment through e-mail over the Internet. Payments are peer-to-peer and virtually instantaneous with a digitally signed receipt to confirm the transaction. Moreover, with **CyberCash MiniPayments**, a consumer can make small payments as if using coins.

10.2 DigiCash

DigiCash was founded in 1990 by D. Chaum to provide a secure electronic payment mechanisms based on public key cryptography he developed. DigiCash uses electronic cash as the payment method, and uses software implementation called **ecash**, instead of storing the cash in a tamper-resistant chip [AD].

"Ecash" can perform transaction between the bank and the user or between two end users like paper cash. Using "ecash", a user can withdraw electronic cash from the bank and store in his local computer. He can then spend these cash without the need to open an account at the shop or to transmit his credit card number over the Internet. Moreover, just like paper cash, the identity of the payer is anonymous and it is based on the concept of blind signatures [43].

1. *Electronic Cash Withdrawal* - The user $A's$ computer creates the cash sealed with a *digital envelop* and transfers to the bank. Then the bank

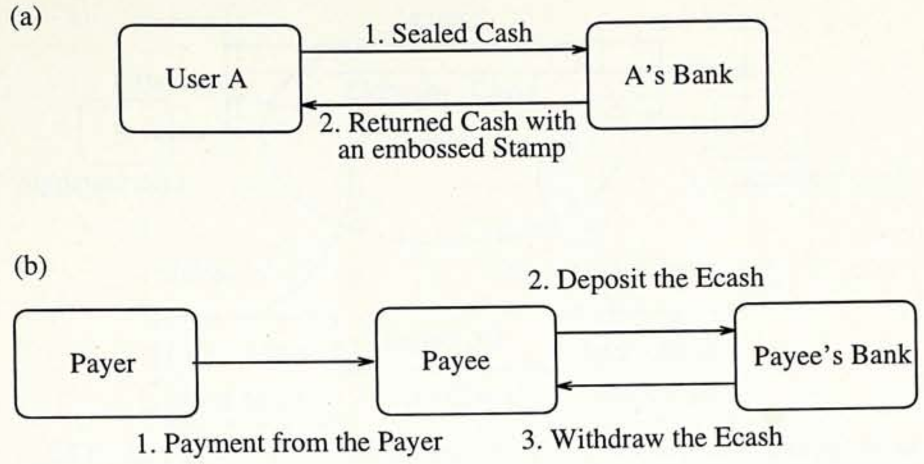


Figure 10.2: Simple Ecash Payment Methods

debits $A's$ account and adds a validating stamp on the envelop and returns it to $A's$ computer. (Figure 10.2a)

2. *Purchase* - After the consumer has made the selection, he transfers the cash to the merchant. The merchant extracts the cash and forwards to the consumer's bank. When the merchant received the validation, he sends the goods or services to the consumer with a receipt.

3. *Person to Person Cash* - When the payee received the cash from the payer, he deposits the cash to the bank. Then he withdraws the cash just deposited after the cash is validated from the payer's bank. (Figure 10.2b)

10.3 The Financial Services Technology Consortium

The Financial Services Technology Consortium (FSTC) is a consortium of financial services providers, national laboratories, universities, and government

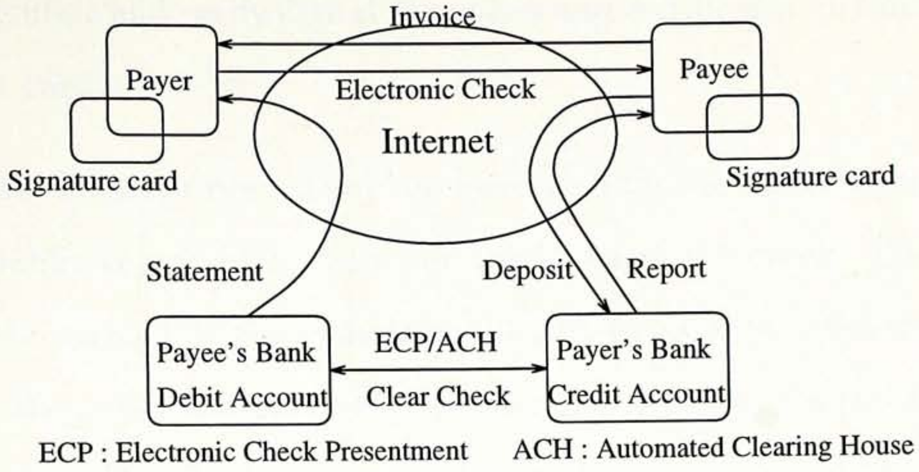


Figure 10.3: FSTC Electronic Check Protocol

agencies who carry out researches and develop interbank technical projects [AE].

10.3.1 Electronic Check Project

The **FSTC Electronic Check Project** models traditional paper check by their *Electronic Check* [AF]. Under their project, (i) paper checkbooks will be replaced by *Electronic Checkbooks*, (ii) pens by signature cards, (iii) signatures by digital signatures and (iv) oridinary mail by e-mail over public network like Internet. The Electronic Check protocol can be simplified as Figure 10.3.

FSTC Electronic Check Component

1. **Electronic Check and Checkbook** - The Electronic Check consists of ascii text block signed by Electronic Checkbook using public key cryptographic signatures. Some possible implementations of Electronic Checkbook include (i) a tamper-resistant PCMCIA card with a mechanism to generate and store unique check identifiers, maintain a check register and

to calculate and verify digital signatures and certificates, (ii) an ISO format smart card.

2. **Digital Signatures** - They are generated by the signer's private key in the public key cryptosystem, and used to sign the check. The check can then be verified by the signer's public key. In order to make the verifier to trust the public key of the signer, the signer can sign the check along with a *a letter of reference* (cryptographic certificate) indicating the signer's name, account number and the signer's public key signed by the signer's bank. In second level, another "letter of reference" is used to indicate the bank's name, details and public key signed by some authority. The verifier can use the public key of the authority to verfiy the bank's and the account certificate and hence verify the signature of the Electronic Check.

3. **Tamper-resistant Signature Cards** - They are used to generate the digital signatures without disclosing the signer's private key. This private key is stored in the processor and the memory on the signature card and cannot be access through the card's connector.

FSTC Electronic Check Technology

1. **Authentication of Electronic Checks** - As stated before, the checks and the checkbooks can be authenticated by public key certificates.

2. **Fraud Prevention** - Digital signature keys are stored in secure signature card to prevent forgery. Hashing function and digital signature are used together to prevent message alternation.

3. **Duplication Prevention** - Duplication is prevented by means of timestamps. Both the Electronic Checks and Electronic Check certificates will expire to prevent them from reuse. The issue bank will also keep track of Electronic Checks which have been presented.

4. **Fraud Deposition Prevention** - The Electronic Check can be encrypted by the intended payee's public key to prevent fraud deposition of the check.

10.3.2 Electronic Commerce Project

FSTC Electronic Commerce Project's aim is to design an infrastructure to support secure and reliable electronic commerce and financial transactions via the existing banking system. The payment model is planned to support other payment mechanisms like FSTC Electronic Cheque and SmartCard based electronic cash [AH].

The payment model can be described by Figure 10.4. Payment instructions are processed by the banks' EPH (*Electronic Payments Handler*), the EPH passes the instructions in appropriate format to a payment processing system. The EPH also receives responses includes: (i) credit card authorization, (ii) money transfer debit advice and (iii) ACH (*Automated Clearing House*) file acceptance or rejection, and forward to the orignator. The EPH consists of four main components:

1. **EPH Frontend** - enforces security for messages to and from the Internet and protected others EPH compoents from the Internet.

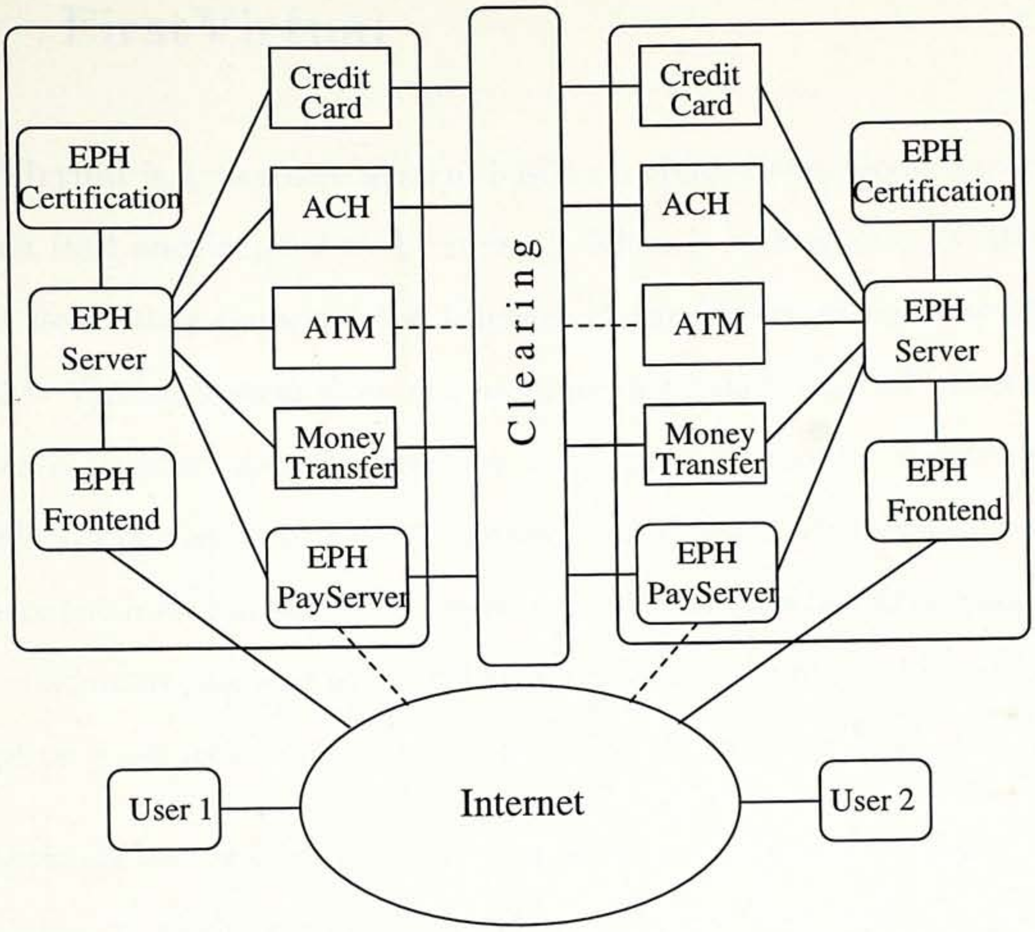


Figure 10.4: FSTC Electronic Commerce Infrastructure

2. **EPH Server** - handles payments protocol, selection of payments mechanisms includes: (i) credit card payments, (ii) ACH, (iii) money transfer, (iv) ATM-like mechanism and (v) EPH-to-EPH clearing mechanism, and interfaces with other payment systems.

3. **EPH Certification Server** - for issuance, validation, revocation of bank cerficates and maintenance of Certificate Revocation Lists.

4. **EPH Payment Server** - supports EPH-to-EPH communication and other payments services.

10.4 FirstVirtual

First Virtual is a payment system based on credit-debit model incorporated in March 1994 and founded by L. Stein, T. Khoury, E. Stefferud, N. Borenstein and M. Rose, they launched the Internet Payment System in October, 1994. The First Virtual system does not use encryption and instead of sending the credit card number over the Internet, the users use account identifiers called **VirtualPIN** to make purchase. Consumers' credit card number and merchants' bank account information will be sent to First Virtual when first signup over a private telephone line and by postal mail respectively and all these information are kept on a secure computer which is not on the Internet [AI].

A VirtualPIN is an alphanumeric character string acts as an alias of financial information or credit card. It identifies the users' and banks' accounts and it is linked to users' financial information off the Internet. First Virtual uses VirtualPIN to provide a secure link between the Internet and the credit cards, banks.

Buying and selling transactions are processed on the Internet whereas banking transaction and financial operations are processed offline. To purchase, the consumer sends his VirtualPIN to the merchant off the Internet. The merchant forwards the VirtualPIN to First Virtual and request to charge the consumer's credit card. First Virtual then asks the consumer for purchase confirmation through e-mail and charges the consumer only when he confirms the transaction by e-mail via a separate, secure line which is not accessible from the Internet.

First Virtual claims that the key authentication mechanism relys on the buyer's e-mail based confirmation of each purchase. If someone tries to use the VirtualPIN to purchase without authorization, the user can inform First Virtual when he received the e-mail for charge confirmation, and the stolen VirtualPIN would be inactivated immediately. To defeat this mechanism, an attacker needs the knowledge of the VirtualPIN, user's e-mail address and the contents of the e-mail which is very unlikely.

First Virtual thinks that using encryption and digital signature will make the system complicated and difficult to use. They suggest that encryption and signature technologies require the use of software and certification infrastructure that are not commonly available and some algorithms are restricted by patents, copyrights and export rules.

10.5 Mondex

The initial concept of Mondex was first invented at NatWest (National Westminster Bank) by T. Jones and G. Higgins in March 1990 and the first patents were applied for in April same year. In 1993, Midland Bank has joined them to be the main partner in UK. In October 1994, they get franchise from Hongkong Bank in the Far East including Hong Kong, China, Singapore and so on. Moreover, the first Mondex service was launched in Swindon, UK in July 1995 [AJ].

Mondex claims that Mondex is electronic cash on a card. They use a smart card to store electronic cash on an encrypted microchip. The card can be used for payment just like using cash and no authorization or signature is required.

They said that the electronic locking system makes it more secure that cash. The smart card provide protection against software or physical attack and re-engineering. Moreover, the chip in Mondex card can generate a unique digital signature which can be recognised by other Mondex card during transaction. This unique signature guarantees the identity of the card and the data transfer is unmodified.

Mondex - Paying on the Internet

According to Mondex, the security depends on the chip on the card but not the network, so money can be transfered over Internet. A Mondex card can handle five different currencies at any time, so an Internet user can make transaction using the currency familiar over the world. The Mondex transaction includes [AK]:

1. Select the goods or services from the merchants that accepts Mondex.
2. Insert the Mondex card into a card reader connected to the computer upon completion of selection.
3. Validate another end of the link and transferred from the card to the merchant's device.
4. Receive the services or goods.

Mondex Devices

1. **Mondex card** - it is an integrated circuit (IC) card, i.e. a plastic card with a small microcomputer chip embedded in it [AL]. This microcomputer was programmed to function as an *electronic purse* which can be loaded with *value* and stored until payment. The balance on the card can be checked by a **balance reader**.

2. **Mondex wallet** - it is a pocket-sized device with a keyboard and a screen, it enables the cardholder to store the value separately and carry less cash on the card. Hence the lost can be minimize even if the card is lost [AL].

3. **Mondex telephone** - it can acts as a personal ATM which allows direct access to a bank account [AL]. A Mondex cardholder can transfer money through a telephone line to other cardholders over the world.

10.6 NetBill

The NetBill project is developed by CMU's (Carnegie Mellon University) Information Networking Institute. The aim of the project is to design protocols and software of Internet commerce. NetBill acts as a third party for authentication, account management, transaction and billing processing [AM].

The NetBill server maintains both the consumers' and the merchants' accounts and linked to conventional financial institutions. A NetBill transaction includes the transfer of information goods, credits and debits of the merchant's and consumer's account respectively [AN]. Moreover, there are client library

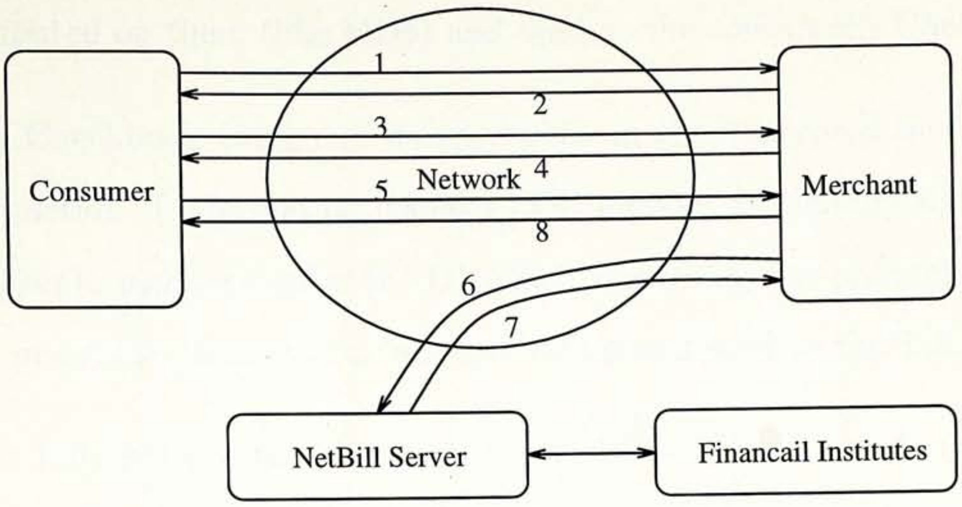


Figure 10.5: NetBill Architecture and Transaction Protocol

called the *CheckBook* and the server library called the *Till* which manage all security and payment protocols, relieving the client and server application. All information exchange between the Checkbook and the Till are encrypted to protect against adversaries.

The NetBill Transaction Protocol can be divided into 8 steps,

1. The consumer requests a price quote through WWW browsers and indicates the Checkbook library. The Checkbook library sends an authenticated request to the Till library and forwards to the merchants' application.

2. The merchant returns the price quote to the consumer's application via the Till and the Checkbook.

3. If the consumer accepts the price quote, the Checkbook then sends a purchase request to the merchant's Till.

4. The Till requests the information from the merchant's application. The information is encrypted by one-time key with the cryptographic checksum

computed on them (like MD5) and send to the consumer's Checkbook.

5. The Checkbook computes its checksum on the encrypted message upon completion of information transfer. The message is digitally signed as the *electronic payment order* (EPO) which indicating the product identifier, the price, checksum and a time-out stamp and send to the Till.

6. The Till compare the checksum of the EPO with its checksum. If they match, the merchant's application appends the decryption key to the EPO and digitally signs the whole message and sends to the NetBill server.

7. The NetBill server verifies the consumer and merchant signatures. If they are valid and the consumer has enough funds in his account, the NetBill server will debit the consumer account and credit the merchant account, log the transaction and make a copy of the decryption key and returns the merchant a digitally signed receipt.

8. The merchant's application forwards this NetBill server's receipt to the Checkbook and complete the transaction.

10.7 NetCash

NetCash is a real-time anonymous electronic payment system on the Internet developed at the Information Sciences Institute of the University (ISI) of the University of Southern California (USC). According to NetCash, the properties of NetCash framework include security, anonymity, scalability, acceptability and interoperability. NetCash uses independently managed and distributed

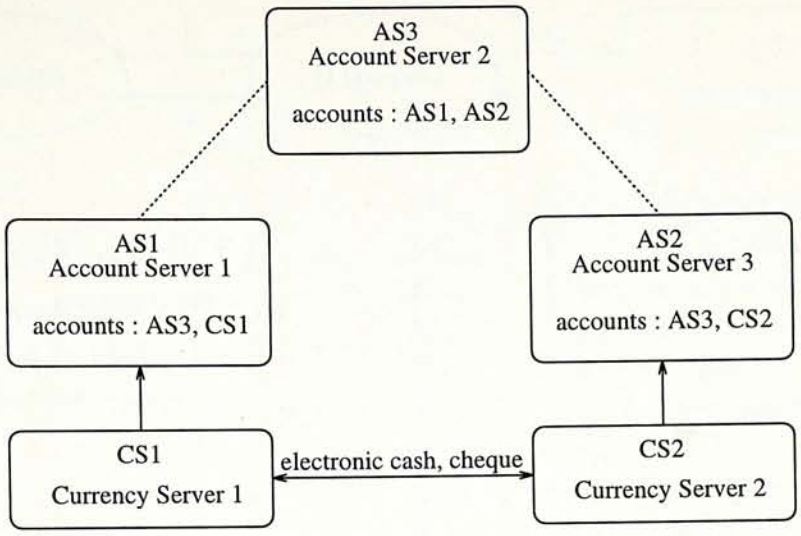
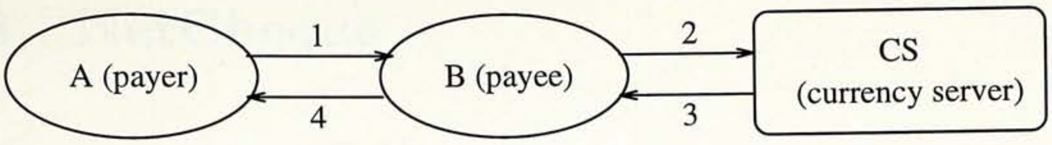


Figure 10.6: An Accounting Heirarchy

currency servers (CS) to allow exchange between electronic cash and other non-anonymous instruments like electronic checks. The users can select the level of anonymity from non-anonymous and weakly anonymous instruments to unconditionally anonymity proposed by D. Chaum [AO, 44, 46].

To setup a currency server, insurance for the new currency is obtained from an insuring agency similar to Federal Deposit Insurance Corporation, the currency is backed by the account balance registered to the CS. The CS obtains certificate of insurance to manage the currency from the insuring agency. An sample accounting heirarchy is shown in Figure 10.6, the **account servers** (AS) maintain accounts for the currency servers and other clients. The CS provides services like: (i) electronic cash verification, (ii) untraceable cash exchange, (iii) using checks to purchase cash and (iv) cashing in cash for checks.

The electronic cash used by NetCash includes a serial number signed with the CS's private key K_{CS}^{-1} as a unique identifier. The CS maintains a list of serial number of all outstanding cash and checks the cash validity based on the



Protocol steps

1. {cash, SK_{AN}, K_{AN}, S_id} K_B
2. {cash, SK_{BN}, transaction} K_{CS}
3. {cash or cheque} SK_{BN}
4. {{amount, T_id, date} K_B^{-1} } SK_{AN}

SK_{AN}	A's freshly generated secret key
SK_{BN}	B's freshly generated secret key
K_{CS}	Currency server's public key
K_B	B's public key
K_B^{-1}	B's private key
K_{AN}	A's freshly generated public session key
S_id	Service identifier
T_id	Unique identifier

Figure 10.7: NetCash Monentary Transactions Protocol

serial number. A simple NetCash protocol that protects the payee from double spending is shown in Figure 10.7.

1. The anonymous payer A sends B the cash, service identifier and two keys SK_{AN} and K_{AN} all encrypted by the payee $B's$ public key K_B.

2. B forwards the received cash to the CS with a newly generated secret key SK_{BN} and the transaction to be performed (a new cash or cheque) and all encrypted by CS's public key.

3. The CS verifys the received cash by checking the serial number against the outstanding list. If the serial number is found, the cash is said valid and not been spent before, then CS sends B the requested instrument (cash or cheque) encrypted by SK_{BN}.

4. B returns a receipt to A which includes the amount paid, a unique identifier T_{id} and date signed by his private key K_B^{-1} and encrypted by A's secret key SK_{AN}. T_{id} is used with the session key K_{AN} to obtain the service.

10.8 NetCheque

NetCheque is another electronic payment system developed at the Information Sciences Institute of the University of Southern California. Users of NetCheque can write electronic checks and spend through e-mail or other network protocols. Signatures are authenticated by Kerberos where reliability and scalability are provided by multiple accounting servers [AP, 45, 46].

NetCheque is a distributed accounting service supporting the credit-debit model. It works like a conventional current account, the account holders digitally sign electronic cheques that include the names of the payer, payee and the finanical institution, the payer's account identifier and the amount. Signed and endorsed cheques are exchanged between accounting servers to settle accounts as shown in Figure 10.6. NetCheque is well suited for frequent micropayment for some commerce, and the performance is obtained through the use of conventional cryptography. NetCheque claims that its use of conventional cryptography makes it faster than systems based on the public key cryptography. Combined with NetCash, the service providers and the users can select different mechanisms based on the level of anonymity requested. NetCash can also use NetCheque to clear payments between currency servers.

NetCheque system uses Kerberos for authentication, the digital signature for signing or endorsing a cheque is a Kerberos ticket called a *proxy* which based on private key cryptography. The information in the cheque includes, (i) the amount of the cheque, (ii) the currency unit, (iii) the expired date, (iv) the charged account, (v) the payee(s) and (vi) the signatures and endorsements verifiable by the accounting server.

When a user signs a cheque, he calls the *write_cheque* function to generate the cheque's cleartext portion (the account, payee, amount and currency unit). Then a Kerberos ticket is obtained for user authentication of accounting server, generation of checksum for the cheque and to append the ticket and the authenticator to the cheque. Finally, the cheque is base 64 encoded and sent to the payee through e-mail or realtime on-line service.

To deposit the cheque, the *deposit_cheque* function is called to read the cleartext part of the cheque, obtains a Kerberos ticket, generates an authenticator to endorse the cheque and appends the endorsement to the cheque. Finally, the endorsed cheque is deposited through an encrypted connection to the payee's account. The authorized users can request for his account balance and other account information through the statement function. It will open an encrypted connection to the accounting server and retrieve the desired information.

Chapter 11

Conclusion

In the second part of the thesis, some security issues in Internet are discussed. In the 90s', the growth of Internet is substantial and now a wide variety of services are available like e-mail, FTP, gopher, TELNET. Moreover, the popularity of World Wide Web provides a better interface betwen the users and the Internet, now the users can access text, audio, image and video information over the Internet. This characteristic provides a new media for business and commerce. However, the openness of Internet also makes Internet vulnerable and security problems will become a great concern for people or companies who want to use Internet for electronic commerce.

Nowadays, many cryptographic algorithms and software packages are available for Internet security. However, the difficulty to understand these algorithms may become a barrier for those who want to use them. So the background knowledge and mathematics behind these security systems are first introduced.

In fact, many security systems rely on one or many private key and public key cryptosystems. A brief introduction to some popular private key (DES, IDEA, RC5) and public key (Diffle-Hellman, RSA and ellptic curves) cryptosystems are given.

Moreover, authentication is also important in Internet, digital signature based on public key is used to solved authenication problems. Hashing function is usually employed together to provide message digest and integrity check. Current digital signature schemes include DSS and Elgamal signature scheme and common hasing functions include MD2 and MD5 are introduced. Some security softwares for Internet like Kerberos, PEM and PGP which based on these algorithms are also discussed.

Having explained the basic building block of an Internet security system, we can now combine them together to construct the infrastructure of electronic commerce system in Internet. The first thing that the consumers can interact with the merchants maybe the Internet browsers. The web browsers provide the user interface for the users to select goods, services and obtain information from the merchants. Meanwhile, the users can send their feedbacks and purchase orders over the Internet to the merchants. The second thing, the processes of payment methods including credit card, electronic cash and electronic cheque are presented. The last thing introduced is some commercial systems for electronic commerce like CyberCash, DigiCash, FirstVirtual, FSTC Project, Mondex, NetBill, NetCash and NetCheque. Each system may use different payment mechanisms and architectures and they are briefly described.

Appendix A

An Essay on Chinese Remainder Theorem and RSA

射鵰英雄傳與萬聯網，有乜關係？　　郁知行著作（指導教授 ：魏克為教授）

Internet （萬聯網）不知不覺間在這一兩年已開始在香港流行起來，不少電腦愛好者都紛紛上網，而一間間的 Internet Service Provider 亦投入市場。 一般來說，最常用的服務都可算是 World-Wide Web (WWW)，要閱覽 WWW 上的資料，用戶需要一個瀏覽器 (browser)。 而在芸芸的瀏覽器之中，相信最被廣泛使用的都可算是 Netscape （網景）。你有否利用過 Internet 去傳遞一些資料譬如是電話、地址甚至是信用卡密碼？如果有的話，你又有否擔心過這些資料會被第三者知道甚至是更改呢？ 因為 Internet 是由世界各地的網絡聯結而成的，所以資料的傳送往往會經過很多不同的國家和不同的網絡，因此 Internet 需要一個十分安全的系統去提供資料保密 (Data Security)。此類系統首先將資料入密(Encryption)，即是將資料轉成一些不能被第三者明白的形式；至於將入密的資料還原的步驟則叫解密 (Decryption)。 而以Netscape 為例，它所採用的保安系統主要由 RSA Data Security,Inc. 所提供。或者你會覺得這些系統的原理一定十分複雜，要涉及很高深的數學運算；事實上，它最基本的原理之一就是小學學過之輾轉相除法。 本文將會嘗試說明在 Internet 的保安系統背後一些簡單的原理。

首先介紹的是輾轉相除法(Euclidean Algorithm)，這是用於計算兩個整數之間的最大公約數(Greatest Common Divisor - GCD)。 此外亦可以用作解整數方程 ax + by≡ gcd(a,b)。 例如要求 144 和 42 之間的最大公約數，

```
  ----------------------
3 )    144    |    42    ( 2
   42×3=126  | 18×2=36
    -------------------
 3 )    18    |     6
     6×3=18  |
    ----------
         0
```

所以 gcd(144,42) = 6。

輾轉相除法有許多應用，包括(1)解孫子點兵法，(2)解射鵰英雄傳中瑛姑之第三道難題及(3)萬聯網上資料保密。 首先講的是孫子點兵法 (Chinese Remainder Theorem)，由中國著名的古代兵法家孫子用於點算軍中士兵的數目。 這問題亦可寫成：求 x 解方程組，

x 除 p1 餘 a1　　　　$x \equiv a1 \pmod{p1}$

x 除 p2 餘 a2　或　$x \equiv a2 \pmod{p2}$

x 除 p3 餘 a3　　　　$x \equiv a3 \pmod{p3}$

而 p1,p2 和 p3 滿足 gcd(p1,p2) = gcd(p2,p3) = gcd(p1,p3) = 1。 之前說過，這算法亦曾在金庸筆下著名武俠小說射鵰英雄傳引用過，當時黃蓉便使用孫子點兵法去解答瑛姑三道難題中的最後一道。

"「今有物不知其數，三三數之賸二，五五數之賸三，七七數之賸二，問物幾何？我知道這是二十三，不過那是硬湊出來的，要列一個每數皆可通用的算式，卻是想破了腦袋也想

不出。」

黃蓉笑道：「這容易得緊。以三三數之，餘數乘以七十；五五數之，餘數乘二十一；七七數之，餘數乘十五。三者相加，如不大於一百零五，即為答數；否則須減去一百零五或其倍數。」

跟著黃蓉便作一首詩以方便記億。「三人同行七十稀，五樹梅花廿一枝，七子團圓正半月，餘百零五便得知。」" < 金庸 - 射鵰英雄傳 > (註)

這道問題可用以下的方程組代表：

x 除 3 餘 2， x 除 5 餘 3， x 除 7 餘 2

用孫子點兵法解此方程組可得，

x 除 (3×5×7)餘 70a + 21b + 15c
或 x 除 105 餘 140 + 63 + 30
或 x 除 105 餘 23

再舉一例，現假設有以下一方程組，

x 除 2 餘 1， x 除 5 餘 4， x 除 9 餘 5

再使用孫子點兵法解此方程組可得，

x 除 (2×5×9)餘 45a + 36b + 10c
或 x 除 90 餘 45 + 144 + 50
或 x 除 90 餘 59

筆者亦嘗試作了一首打油詩。「雙龍拱照九五尊，五行幻變六六順，九星聯珠方十日，九句盡去謎揭盅。」

以上這些定理現在都已被廣泛利用在各種不同的 Internet 保安系統之上。

前面曾提及過 Netscape 所使用的保安系統是由 RSA Data Security Inc.所提供的。其中的 RSA 系統是由美國的麻省理工(MIT)的 Ron Rivest，Adi Shamir 和 Leonard Adleman 於1978年最先提出的。 這是現時其中一個最可靠的公開鑰密碼系統(Public Key Cryptosystem)。 公開鑰密碼系統不同於傳統的(秘密鑰 Private Key)密碼系統。 使用傳統的密碼系統就好像我們日常使用的鑰匙(key)，我們用相同的鑰匙來上鎖(入密)和開鎖(解密)。 假設 A 想將一明文(Plaintext)傳送給 B， A 便使用秘密鑰將明文入密以得出密文(Ciphertext)。 當 B 收到密文之後，他必須使用相同的秘密鑰將密文解密。 因此 B 必須預先將秘密鑰傳送給 A，但若果在傳送過程被第三者知道的話，那麼這第三者便可將 A 傳送給 B 的密文解密，這便是密鑰的分配問題。 除此之外，秘密鑰密碼系統亦難以用於確認用戶的身份。

公開鑰密碼系統便可解決這些問題，使用公開鑰密碼系統，每個用戶都有一組公開鑰(Public Key) 和一組秘密鑰 (Private Key)。 公開鑰只是用作上鎖(入密)，而秘密鑰只是用作開鎖(解密)。 用戶會將公開鑰公開給其他用戶查看，而自己則將秘密鑰保密，因此第三者便不能進行解密，此外公開鑰的設計亦會令第三者極難(幾乎不可能)由公開鑰求出秘密鑰。 假設 A 想將一明文傳送給 B， A 便使用 B 的公開鑰將明文入密以得出密文。

當 B 收到密文之後，他便使用自己的秘密鑰將密文解密。

RSA 的兩個公開鑰的產生過程如下：選擇兩個質數 p 和 q，設 p = 3 和 q = 5。計算 n = p×q = 15。 隨機選擇一個入密密鑰 e 令 gcd(e，(p-1)×(q-1)) = 1。 現選 e = 11，這樣這系統的兩個公開密鑰便是 n = 15 和 e = 11。 跟著使用輾轉相除法去求解密鑰 d 使滿足 e×d 除 8 餘 1 ；得出 d = 3。

跟著RSA 的入密過程如下：設明文 M 為 7，那麼密文 C 即為 M^e (mod n) = 7^11 (mod 15) = 13。 而解密的過程是：明文 M 是 C^d (mod n) = 13^3 (mod 15) = 2197 (mod 15) = 7。 再舉一例：設明文 M 為 3，那麼密文 C = 3^11 (mod 15) = 12。 再使用 d 解密得 M = 12^3 (mod 15) = 1728 (mod 15) = 3。

RSA 的原理是：如 n = p×q，m = (p-1)×(q-1)，而 k 是 m 的倍數。 那麼對任意一個數字 M，M^(k+1) 除 n 餘 M。 例如 n = 2×5 = 10，m = 4，k = 8。 若 M = 3， 那麼 3^(8+1) = 19683，除 10 便餘 3 = M。 又例如 n = 3×7 = 21， m = 12，k = 12。 若 M = 2， 那麼 2^(12+1) = 8192，除 21 便餘 2 = M。 在 RSA 裡，e×d 除 m 餘 1， 所以 e×d = k+1 (k 是 m 的倍數)。 因此解密時， C^d (mod n) = (M^e)^d (mod n) = M^(e×d) (mod n) = M^(k+1) (mod n) = M。

以上所提及過的例子所用的數字都是比較細的，但是在實際用途上這些數字都是十分大。 因為破解 RSA 的最明顯方法是分解(factorize) n，n 被分解之後便可知 p 和 q，跟著解密鑰 d 亦可被計算出來。 所以如果 n 越大的話，代表 n 越難被分解，一般來說，n 的大小需有大約 512 bits (154 digits)便可。

從以上種種例子可見一些看似十分複雜的系統往往都只是依靠一些很簡單的原理，甚至是小學知的簡單理論，也有非常高深的運用。 只要大家有興趣研究一下的話，要明白這些原理也並不困難。

註 ： 跟據 < 辭海 > 孫子數物題，此法又稱鬼谷算，隔牆算，剪管術，秦王暗點兵。明朝程大位算法統宗之中稱之為韓信點兵，程大位並做詩云 「三人同行七十稀，五樹梅花廿一枝 ，七子團圓正半月，除百零五便得知。」

Bibliography

[1] C. Reiners and H. Rohling, "Multicarrier Transmission Technique in Cellular Mobile Communications Systems", *IEEE Vech. Tech.*, pp.1645-1649, 1994.

[2] P. Crespo, M. L. Honig and J. A. Salehi, "Spread-Time Code Division Multiple Access", *IEEE Global Telecommunications Conference*, pp.836-840, 1991.

[3] G. Fettweis, A.S. Bahai and K. Anvari, "On Multi-Carrier Code Division Multiple Access (MC-CDMA) Modem Design", *IEEE Vech. Tech.*, pp.1670-1674, 1994.

[4] S. Kaiser, "OFDM-CDMA versus DS-CDMA: Performance Evaluation for Fading Channels", *IEEE ICC* pp. 1722-1726, 1995.

[5] H. Sari, G. Karam and I. Jeanclaude, "An Analysis of Orthogonal Frequency-Division Multiplexing for Mobile Radio Applications", *IEEE Vech. Tech.*, pp.1635-1639, 1994.

[6] F. Classen and H. Meyr, "Frequency Synchronization Algorithms for OFDM Systems suitable for Communication over Frequency Selective Fading Channels", *IEEE Vech. Tech.*, pp.1655-1659, 1994.

[7] J.A.C. Bingham, "Multicarrier Modulation for Data Transmission: An Idea Whose Time has Come", *IEEE Comm. Magazine*, pp.5-14, May. 1990.

[8] R. Kohno, R. Meidan and L. B. Milstein, "Spread Spectrum Access Methods for Wireless Communications", *IEEE Comm. Magazine*, pp.58-67, Jan. 1995.

[9] A. J. Viterbi, "The Orthogonal-Random Waveform Dichotomy for Digital Mobile Personal Communications", *IEEE Personal Comm.*, pp.18-24, 1994

[10] J. G. Proakis, "Digital Communications" *McGraw Hill* 1989.

[11] E. A. Lee and D. G. Messerschmitt, "Digital Communication" *Kluwer AP* 1988.

[12] L. W. Couch II, "Digital and Analog Communication Systems", *Macmillan*, 1993.

[13] B. K. Levitt, "Spread Spectrum Communications Volume III", *Computer Science Press*, 1985.

[14] B.M. Popović, "Synthesis of Power Efficient Multitone Signals with Flat Amplitude Spectrum", *IEEE Trans. Commun.*, vol 39, pp.1031-33, July 1991.

[15] M.J.E. Golay, "Complementary Series", *IRE Trans. Inform. Theory*, vol. IT-7, pp.82-87, Apr. 1961.

[16] S.U.Zaman and K.W. Yates, "Multitone Synchronization for Fading Channels", *IEEE ICC*, pp.946-949, 1994.

[17] J. Singer, "Elements of Numerical Analysis", *Academic Press*, Chapter 9 , 1964 .

[18] B. Schneier, "Applied Cryptography: Protocols, Algorithms, and Source Code in C", *Wiley*, 1994.

[19] D. R. Stinson, "Cryptography: Theory and Practice", *CRC Press*, 1995.

[20] A. S. Tanenbaum, "Computer Networks", *Prentice Hall*, 1989.

[21] D. E. Comer, "Internetworking with TCP/IP Vol. I.", *Prentice Hall*, 1991.

[22] R. Oppliger, "Internet security enters the Middle Ages", *Computer*, p.100-101, Oct. 1995.

[23] P. Ribenboim, "The Book Of Prime Number Records", *Springer-Verlag*, 1989.

[24] P. Giblin, "Primes and Programming", *Cambridge University Press*, 1993.

[25] T. M. Cover and J. A. Thomas, "Elements of Information Theory", *Wiley*, 1991.

[26] H. Niederreiter, "Random Number Generation and Quasi-Monte Carlo Methods", *Capital City Press*, 1992.

[27] Eastlake, Crocker and Schiller, "Request for Comments :1750 - Randomness Recommendations for Security".

[28] C. H. Meyer and S. M. Matyas, "Cryptography: A New Dimension in Computer Data Security", *John Wiley & Sons*, 1982.

[29] H. C. A. V. Tilborg, "An Introduction to Cryptology", *Kluwer Academic Publishers*, 1993.

[30] R. Zimmermann, A. Curiger, H. Bonnenberg, H. Kaeslin, N. Felber and W. Fichtner "A 177Mb/s VLSI Implementation of the International Data Encryption Algorithm" *IEEE Solid-State Circuits*, 1994.

[31] E. Dawson, "Linear Feedback Shift Registers and Stream Ciphers" in "Number Theory and Cryptography", *Cambridge University Press*, 1991.

[32] W. Diffie and M. E. Hellman, "New Directions in Cryptography" in "Secure Communications and Asymmetric Cryptosystems" *AAAS Selected Symposium*, 1982.

[33] R. C. Merkle, "Protocols for Public Key Cryptosystems" in "Secure Communications and Asymmetric Cryptosystems" *AAAS Selected Symposium*, 1982.

[34] R. C. Merkle and M. E. Hellman, "Hiding Information and Signatures in Trapdoor Knapsacks" in "Secure Communications and Asymmetric Cryptosystems" *AAAS Selected Symposium*, 1982.

[35] R. L. Rivest, A. Shamir and L. M. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems" in "Secure Communications and Asymmetric Cryptosystems" *AAAS Selected Symposium*, 1982.

[36] A. J. Menezes, "Elliptic Curve Public Key Cryptosystems", *Kluwer Academic Publishers*, 1993.

[37] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems", *Advances in Cryptology - CRYPTO'90*, pp.2-21, *Springer-Verlag*,1990.

[38] E. Biham and A. Shamir, "Differential Cryptanalysis of the full 16-round DES", *Advances in Cryptology - CRYPTO'92*, pp.487-496, *Springer-Verlag*, 1992.

[39] S. Garfinkel and G. Spafford, "Practical UNIX Security", *O'Reilly & Associates, Inc.*, 1991.

[40] S. M. Bellovin and W. R. Cheswick, "Network Firewalls", *IEEE Commun. Magazine*, p.50-57, Sept. 1994.

[41] P. Zimmermann, "Pretty Good Privacy - Public Key Encryption for the Masses, Volume I: Essential Topics", *PGP User's Guide.*

[42] D. C. Little, "Commerce on the Internet", *IEEE Multimedia*, p.74-78, 1994.

[43] D. Chaum, "Blind Signatures for Untraceable Payments", *Advances in Cryptology - CRYPTO'82*, pp.199-204, 1982.

[44] G. Medvinsky and B. C. Neuman "NetCash: A design for practical electronic currency on the Internet" *ACM Conf. on Comp. and Comm. Security*, 1993.

[45] B. C. Neuman and G. Medvinsky "Requirements for Network Payment: The NetCheque Perspective", *IEEE Compcon '95*, 1995.

[46] G. Medvinsky and B. C. Neuman "Electronic Currency for the Internet", *Electronic Markets*, pp.23-24, Oct. 1993.

URL References

[A] ”Request for Comments :1319 - The MD2 Message-Digest Algorithm” *http://www.cis. ohio-state.edu/htbin/rfc/rfc1319.html.*

[B] ”Request for Comments :1321 - The MD5 Message-Digest Algorithm” *http://www.cis. ohio-state.edu/htbin/rfc/rfc1321.html.*

[C] ”Cryptographic Random Numbers” *http://www.clark.net/pub/cme/ P1363/ranno.html.*

[D] ”$IDEA^{TM}$ - the International Data Encryption Algorithm from ascom” *http:// www.ascom.ch/Web/systec/security/idea.htm.*

[E] ”Alliance With Ascom Systec to Commercially License IDEA encryption algorithm” *http://www.ascom.ch/Web/systec/press/release.htm.*

[F] R. L. Rivest, ”The RC5 Encryption Algorithm”, *http://theory.lcs.mit.edu/ rivest/publications.html.*

[G] ”RSA’s Frequently Asked Questions About Today’s Cryptography - Miscellaneous” *http://www.rsa.com/rsalabs/faq/faq_misc.html.*

[H] ”Community ConneXion - Hack Netscape” *http://www.c2.org/ hacknetscape.*

[I] I. Goldberg and D. Wagner ”Netscape SSL implementation cracked!” *http://hplyot. obspm.fr/ dl/netscapesec.*

[J] ”Potential Vulnerability in Netscape Products” *http:// www.netscape.com/newsref/std/random_seed_security.html.*

[K] P. C. Kocher "Cryptanalysis of Diffie-Hellman, RSA, DSS, and Other Systems Using Timing Attacks" *http://www.cryptography.com/ timingattack.html.*

[L] "Security Dynamics" *http://www.securid.com/ID224.346452114269/.*

[M] "The Smart Card Forum" *http://www.smartcrd.com.*

[N] "Keeping Your Site Comfortably Secure : An Introduction to Internet Firewalls", *http://csrc.ncsl.nist.gov/nistpubs/800-10.*

[O] "The Kerberos Network Authentication Service (V)", *http:// www.cis.ohio-state.edu/htbin/rfc/rfc1510.html.*

[P] "Privacy Enhancement for Internet Electronic Mail, Part I : Message Encryption and Authentication Procedure", *http:// andrew2.andrew.cmu.edu/rfc/rfc1421.html.*

[Q] "Privacy Enhancement for Internet Electronic Mail, Part III : Algorithms, Modes, and Identifiers", *http://andrew2.andrew.cmu.edu/rfc /rfc1423.html.*

[R] "Secure NCSA Mosaic Home Page" *http://www.commerce.net/software /SMosaic/Docs/SMosaic.home.html.*

[S] "The Secure HyperText Transfer Protocol", *http:// www.commerce.net:80/information/standards/drafts/shttp.txt.*

[T] "Welcome to Netscape" *http://home.netscape.com.*

[U] "On Internet Security" *http://home.netscape.com/info/security-doc.html.*

[V] "Netscape Data Security" *http://home.netscape.com/newsref/ref/netscape-security.html.*

[W] "The SSL Protocol" *http://home.netscape.com/newsref/std/SSL.html.*

[X] "The SSL Protocol Version 3.0" *htpp://home.netscape.com/eng/ssl3/ssl-toc.html.*

[Y] "The $HOTJAVA^{TM}$ Browser : A White Paper" *http://java.sun.com /1.0alpha3/doc/overview/hotjava/index.html.*

[Z] "The $Java^{TM}$ Language : A White Paper" *http://java.sun.com /1.0alpha3/doc/overview/java/index.html.*

[AA] "HotJava(tm) : The Security Story" *http://java.sun.com/1.0alpha3/doc/security/security.html.*

[AB] "CyberCash Home Page" *http://www.cybercash.com/cybercash*

[AC] "Secure Internet Credit Card Payment" *http://www.cybercash.com /cybercash/who-we-are/sixsteps.html*

[AD] "DigiCash home page" *http://www.digicash.com*

[AE] "FSTC Home Page" *http://www.fstc.org/index.html.*

[AF] "FSTC Electronic Check Project" *http://www.fstc.org/projects/echeck /index.shtml.*

[AG] "FSTC Electronic Check Project Details" *http://www.fstc.org/projects /echeck/echeck2.shtml.*

[AH] "Electronic Payments Infrastructure: Design Considerations" *http://www.fstc.org/projects/commerce/public/epaydes.htm.*

[AI] "FV: Home Page" *http://www.fv.com/html/fv_main.html*

[AJ] "Mondex Home Page" *http://www.mondex.com/mondex/home.htm.*

[AK] "Paying on the Internet (Mondex)" *http://www.mondex.com/mondex/inter.htm.*

[AL] "Mondex Devices" *http://www.mondex.com/mondex/device.htm.*

[AM] "NetBill Project Home Page", *http://www.ini.cmu.edu/NETBILL/home.html.*

[AN] M. Sirbu and J. D. Tygar, "NetBill: An Internet Commerce System Optimized for Network-Delivered Services", *IEEE Personal Comm.*, Aug. 1995.

[AO] "The NetCash(SM) anonymous network payment system" *http://gost.isi.edu:80/info/netcash.*

[AP] "The NetCheque(SM) network payment system", *http://gost.isi.edu:80/info/netcheque.*