韓國 軍事運營分析 學會誌
第12卷 第2號, 1986.12.

資料傳送 保安을 위한 暗號化시스템 設計에 관한 硏究

禹 鍾 植*

Abstract

Data Security in computer communication is becoming serious problem with developing computer network.

This paper intended to design an encryption system to improve processing speed and to get higher degree of security by combining random number generation method, substitution method and Vernam's encryption method.

1. 序 論

最近 데이타通信과 데이타베이스에 관한 技術이 급격히 發展함에 따라 여러 使用者가 資源을 共有하게 되고 資料의 移動이 빈번하게 되어 情報가 外部로 노출될 수 있는 可能性이 높아졌다.

그런데 電算機를 利用한 데이타通信은 秘密資料를 送受信할 경우에 資料의 변조 및 도난을 막기 위한 對策이 마련되어야 한다. 그래서 秘密資料를 送受信할 수 있는 秘話機의 必要性이 인식되고 있으나 國內生産은 어려운 실정이며 外國에서 生産되고 있는 몇가지 秘話機가 있으나 그들의 獲得이 어려울 뿐만 아니라 獲得한다 하더라도 外國의 技術로 제작된 秘話機를 使用한다는 것은 秘話機의 특성을 충분히 보장받을 수 없다. 따라서 本 論文은 현재 우리가 이용하고 있는 소프트웨어的인 暗號化方法을 基準으로 보다 秘度가 높고 처리속도도

* 國防大學院

빠르며 운영유지가 간편한 暗號化시스템을 具現하는데 目的을 두었다. 이러한 暗號化시스템을 具現하기 위해서 지금까지 公開되어 일반화된 暗號化技法인 代置法, 난수(random number)에 의한 技法, 中央統制型 키이(key) 管理技法, VERNAM의 暗號化技法 등을 응용하여 暗號化시스템을 設計하고 評價를 實施하였다.

2. 暗號化技法의 一般的 考察

가. 暗號化 概要

資料保安이란 非認可된 사용, 수정, 파괴 등의 행위로부터 資料를 보호하는 것으로서 資料의 사용을 제한하는 使用制限(access control)方法, 사용자의 水準에 맞는 部分만을 許用하는 多段階保安(multilevel security) 方法이 있는데 위의 方法들은 電算機 內部에 있는 資料에 대한 保安方法이고 資料가 外部로 送受信될 경우에는 資料를 暗號化해야 한다

暗號化시스템은 平文, 키이(key), 알고리즘, 暗號文으로 구성되며 平文은 暗號化되지 않은 원래의 文章이고, 暗號文은 暗號化過程을 거쳐 변형된 文章이다. 이러한 暗號化過程에서 사용되는 변수가 키이이다.

平文이 暗號文으로 변형되는 過程을 暗號化過程(encryption)이라 하고 이와 반대로 暗號文을 원래의 平文으로 만드는 過程을 解讀過程(decryption)이라 하며 이들 두 過程을 要約하면 〈그림 2-1〉과 같다.

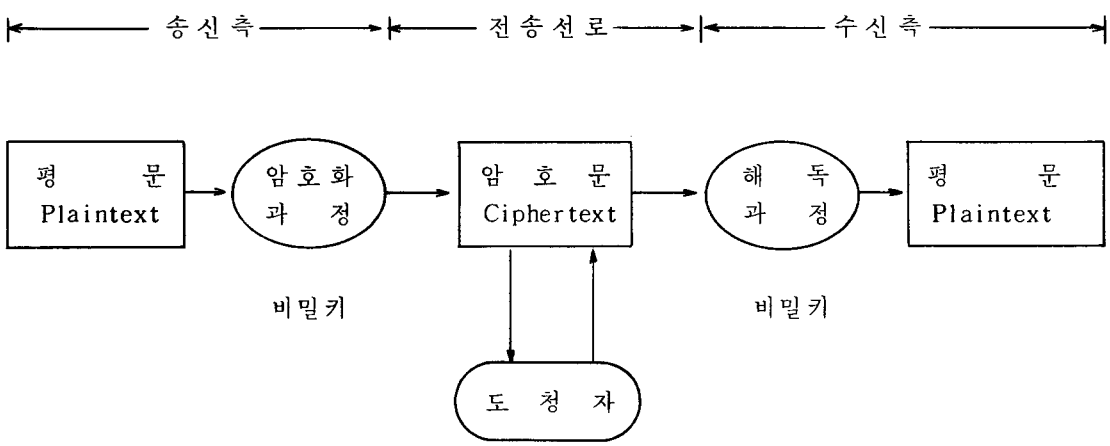


그림 2-1. 일반적인 暗號化 모델

나. 소프트웨어的인 暗號化技法

(1) 位置交換法(Transposition Method)

이 方法은 平文에서 사용된 모든 文字를 그대로 사용하면서 단지 位置만을 변경시켜 본래의 뜻을 알아볼 수 없게 만드는 方法이다. 이 方法은 電算機에 의해 쉽게 解讀될 수 있기 때문에 별로 사용하지 않는다. 왜냐하면 平文의 文字가 暗號文에 그대로 사용되므로 暗號文과 平文의 文字頻度數가 같기 때문에 位置交換法에 의한 暗號化技法이라는 사실이 쉽게 노출

될 수 있다.

다음의 "보기"들은 전형적인 位置交換法이다.

보기 1 平文 전체를 逆順으로 나열한 후 다섯文字씩 짝을 짓는 方法.

平 文 : DATA SECURITY AND ENCRYPTION

暗號文 : NOITP YRCNE DNAYT IRUCE SATAD

보기 2 平文을 각 單語別로 逆順으로 나열한 후 다섯文字씩 짝을 짓는 方法

平 文 : DATA SECURITY AND ENCRYPTION

單語別 逆順 : ATAD YTIRUCES DNA NOITPYRCNE

暗號文 : ATADY TIRUC ESDNA NOITP YRCNE

보기 3 平文을 다섯文字씩 (공백을 포함하여) 짝을 지은 후 각각에 대하여 位置를 변형시키는 方法

平 文 : DATA AND FILE OF COMPUTER

위치변경순서 : $\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 5 & 3 & 2 \end{bmatrix}$

暗號文 : AD TA AFDN IDELOFMC EPRTU

보기 4 MATRIX를 이용하여 行 (row) 으로 채워넣고 列 (column) 로 선택하여 다섯文字씩 짝을 짓는 方法.

平 文 : DATA SECURITY AND ENCRYPTION

D	A	T	A	S
E	C	U	R	I
T	Y	A	N	D
E	N	C	R	Y
P	T	I	O	N

暗號文 : DETEP ACYNT TUACI ARNRO SIDYN

"보기 4 "와 같은 방법은 메트릭스안에 文字를 채워넣는 방법에 변화를 줌으로써 다양하게 사용할 수 있다. (예를 들면 時計方向 나열, "ㄹ"자 모양 나열 등등이 있다)

(2) 代置法 (Substitution Method)

代置法은 平文의 文字나 文字의 집합이 다른 文字나 文字의 집합으로 代置되는 것으로서 平文의 원래 順序는 유지하되 각각의 文字를 변형시키는 方法이다.

(가) 單一文字 代置法

이 方法은 平文의 각 文字나 記號를 다른 文字나 記號로 代置하는 것으로서 키이의 文字들을 入力과 出力으로 나누어서 入力에 해당하는 出力 文字를 하나씩 대응시켜 暗號文을 만드는 方法이다.

〈보 기〉

平 文 : DATA SECURITY AND ENCRYPTION

入力키이 : ㅎABCDEFGHIJKLMNOPQRSTUVWXYZ

出力키이 : KELQSXZDGMARPFJTWㅎUNHBOVYIC

暗號文 : SEHSKNXQBUMHIKEJSKXJQUIWHMTJ

이 方法은 使用可能한 키이가 위의 경우에 $26! = 4 \times 10^{26}$ 개이므로 키이를 알지 못하고는 解讀이 어려울듯 보이지만 暗號文을 盜聽해서 각 文字의 出現頻度數를 찾아내 비교적 쉽게 解讀되는 方法이다.

(나) 複數文字 代置法

單一文字 代置法의 短點을 補完하기 위해서 暗號文의 각 文字 出現頻度數를 위장시킨 것이 複數文字 代置法이다.

이 方法은 平文의 각 文字가 키워드(keyword)에 의해서 하나 以上의 文字로 代置되어 頻度數 分析으로는 解讀하기 어렵다.

〈보 기〉

平 文 : WE HAVE A NEW DEVICE

키 이 : SECURITY

평문 키이			암호문
↓	↓		↓
W	S	= 23+19 (MOD27) = 15 =	O
E	E	= 5+5 (MOD27) = 10 =	J
	C	= 0+3 (MOD27) = 3 =	C
H	U	= 8+21 (MOD27) = 2 =	B
A	R	= 1+18 (MOD27) = 19 =	S
V	I	= 22+9 (MOD27) = 4 =	D
E	T	= 5+20 (MOD27) = 25 =	Y
	Y	= 0+25 (MOD27) = 25 =	Y
A	S	= 1+19 (MOD27) = 20 =	T
	E	= 0+5 (MOD27) = 5 =	E
N	C	= 14+3 (MOD27) = 17 =	Q
E	U	= 5+21 (MOD27) = 26 =	Z
W	R	= 23+18 (MOD27) = 14 =	N
	I	= 0+9 (MOD27) = 9 =	I
D	T	= 4+20 (MOD27) = 24 =	X
E	Y	= 5+25 (MOD27) = 3 =	C
V	S	= 22+19 (MOD27) = 14 =	N
I	E	= 9+5 (MOD27) = 14 =	N
C	C	= 3+3 (MOD27) = 6 =	F
E	U	= 5+21 (MOD27) = 26 =	Z

暗號文 : OJCBSDYYTEQZNIXCNNFZ

이 方法은 1) 키워드의 길이가 平文보다 길거나, 2) 키이로서 한번만 사용될 때, 3) 어떤 平文도 盜聽者에게 알려지지 않았을 때 解讀되기 어려운 方法이다.

(3) 代數的인 方法

代數的인 方法은 平文의 文字를 미리 定義된 숫자로 바꾸고 이 숫자를 數學的으로 처리한 다음에 다시 이것을 文字로 바꾸는 方法이다. 또 다른 方法은 平文의 文字를 2進化10進數(BCD : Binary Coded Decimal)로 표시한 다음에 論理 혹은 산술演習을 行하여 暗號化하는 方法이다. 이러한 代數的인 暗號化 方法은 컴퓨터와 關聯되어 가장 많이 사용되는 方法이다.

(가) VERNAM의 暗號化方法

이 方法은 본래 텔레타이프에 이용하기 위해서 平文과 키이의 BCD코드를 EX-OR연산을 해서 暗號文을 만드는 方法이다. 예를 들어 BCD코드로 平文이 010001(A)이고, 키이가 110111(Z)이면 다음과 같이 된다.

平 文 : 010001 (A)

키 이 : 110111 (Z)

暗號文 : 100110 (K)

이 方法은 暗號化過程과 解讀過程이 정반대이기 때문에 暗號文과 키이를 다시 EX-OR연산을 行하면 원래의 平文을 얻을 수 있다.

(나) 連續方程式에 의한 暗號化方法

이 方法은 임의의 숫자화한 알파벳을 이용하여 4개의 文字를 한 그룹으로 暗號化하는 方法이다.

(임의의 숫자화한 알파벳)

A	B	C	D	E	F	G	H	I	J	K	L	M
4	8	25	2	9	20	16	5	17	3	0	22	13
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
24	6	21	15	23	19	12	7	11	18	1	14	10

다음의 方程式에서 X_i는 平文의 글자를 Y_i는 暗號文의 글자를 나타낸다.

暗號化 方程式 : (mod 27 사용)

$$Y_1 = 8X_1 + 6X_2 + 9X_3 + 5X_4$$

$$Y_2 = 6X_1 + 9X_2 + 5X_3 + 10X_4$$

$$Y_3 = 5X_1 + 8X_2 + 4X_3 + 9X_4$$

$Y_4 = 10X_1 + 6X_2 + 11X_3 + 4X_4$

解讀 方程式 : (mod27 사용)

$X_1 = 23Y_1 + 20Y_2 + 5Y_3 + 1Y_4$

$X_2 = 2Y_1 + 11Y_2 + 18Y_3 + 1Y_4$

$X_3 = 2Y_1 + 20Y_2 + 6Y_3 + 25Y_4$

$X_4 = 25Y_1 + 2Y_2 + 22Y_3 + 25Y_4$

예를 들어 " HELP " 라는 平文을 暗號化하는 過程을 보면 임의의 숫자화한 알파벳에 의해 $X_1 = 5$ (H), $X_2 = 9$ (E), $X_3 = 22$ (L), $X_4 = 21$ (P)가 되고 이것을 暗號化 方程式에 代入하여 계산하면 $Y_1 = 7$, $Y_2 = 15$, $Y_3 = 10$, $Y_4 = 14$가 된다. 여기서 7, 15, 10, 14는 임의의 숫자화한 알파벳에 의해 각각 U, Q, Z, Y이므로 "HELP"의 暗號文은 "UQZY"가 된다.

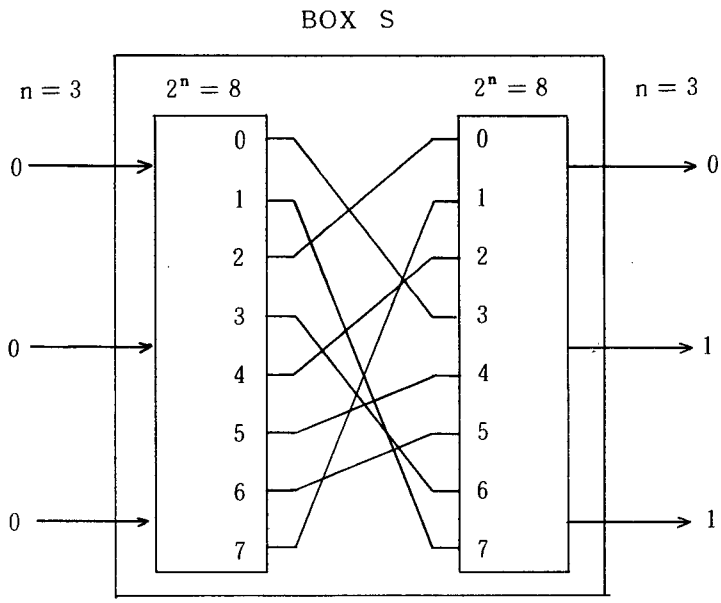
다. 하드웨어的인 暗號化技法

(1) LUCIFER技法

代置法이나 位置交換法으로 각각 暗號化하는 것보다 두가지 技法을 組合해서 暗號化하면 解讀이 더욱 어려워진다. 이러한 原理를 이용한 方法이 合成法인데 代表的인 형태가 LUCIFER技法이다.

이 方法은 代置回路 (S-BOX)와 順列回路 (P-BOX)로 구성되어 있으며 代置回路는 〈그림 2-2〉와 같이 平文의 n비트를 暗號文의 n비트로 바꾸는 裝置로서 n가 크면 秘度가 더욱 높아진다. 이것은 入力비트를 出力비트로 代置하기 위해 단순히 0과 1을 더하는 형태가 아니라 2개의 스위치에 의해서 n개의 비트를 2^n개의 비트로 다시 이것을 n개의 비트형태로 바꿀 수 있도록 만든 裝置이다.

順列回路는 〈그림 2-3〉과 같으며 平文의 n비트를 暗號文의 n비트로 바꾸어 주는 裝置이다. 이것은 入力이 들어가서 어떻게 出力이 나오는가를 확인하지 않고는 解讀이 不可



IN	OUT
0 0 0	0 1 1
0 0 1	1 1 1
0 1 0	0 0 0
0 1 1	1 1 0
1 0 0	0 1 0
1 0 1	1 0 0
1 1 0	1 0 1
1 1 1	0 0 1

그림 2-2. 代置回路 (S-BOX)

能하다.

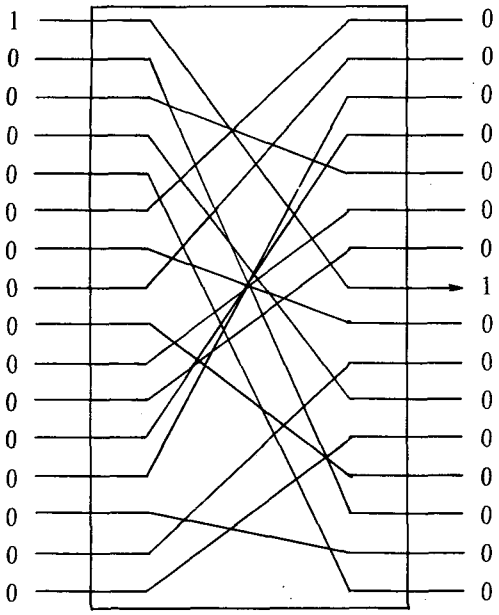


그림 2-3. 順列回路(P-BOX)

이상에서 설명한 代置回路와 順列回路를 組合해서 만든 合成回路가 LUCIFER이다. LUCIFER의 처리속도는 96,970 bps인 반면에 이것을 소프트웨어로 구현할 경우 8,000 bps (CDC6400 컴퓨터 이용)로서 처리속도면에서 하드웨어로 구현하는 方法이 훨씬 빠르다.

(2) DES알고리즘(Data Encryption Standard Algorithm)

DES는 컴퓨터의 데이터를 보호하기 위한 數學的인 알고리즘으로서 BCD데이터를 사용하며 64비트의 情報를 暗號化하기 위하여 64비트의 키이를 이용한다.

DES는 64비트 키이중에서 56비트를 사용하고 나머지 8비트는 패리티(parity)檢査用으로 사용한다. 즉 키이는 8비트씩 나누어지고 그 중 7비트는 알고리즘에 사용되며 8번째 비트는 홀수 패리티를 유지하기 위하여 이용된다. 56비트의 組合方法은 7×10^{16}이상이므로 키이를 알지 못하고 解讀하기는 거의 不可能하다. DES알고리즘의 槪要는 〈그림 2-4〉와 같다.

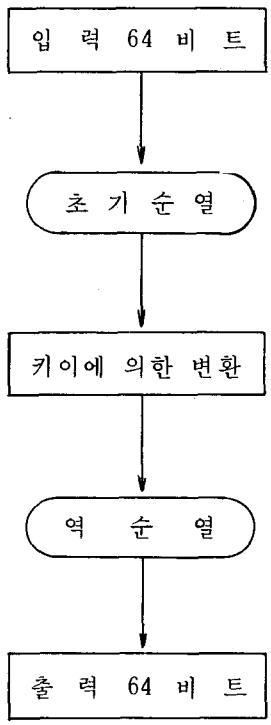


그림 2-4. DES알고리즘의 槪要

수행절차를 보면 최초 入力된 64비트의 각 비트位置를 서로 바꾼다. 이를 初期順列이라고 하며 〈그림 2-5〉와 같다. 다음은 키이에 의한 변환과정을 실시하고 마지막으로 변환이 행해진 결과에 대해 逆順列을 행한다. 이 過程은 初期順列의 逆過程이며 〈그림 2-6〉과 같다.

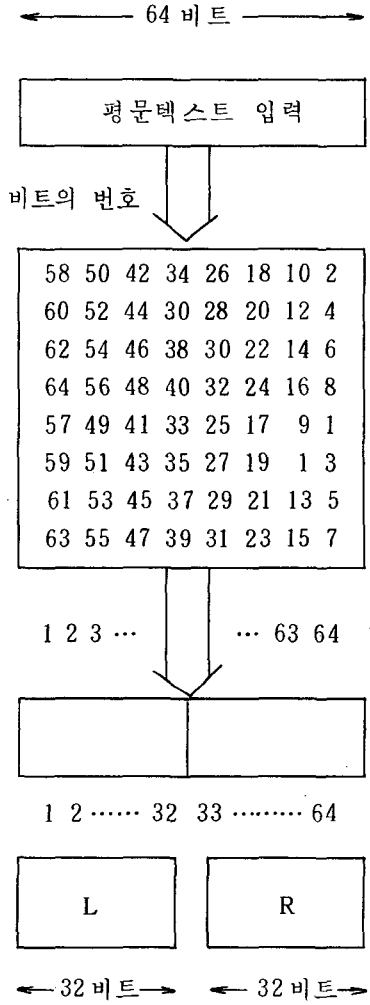


그림 2 - 5 . 初期順列過程

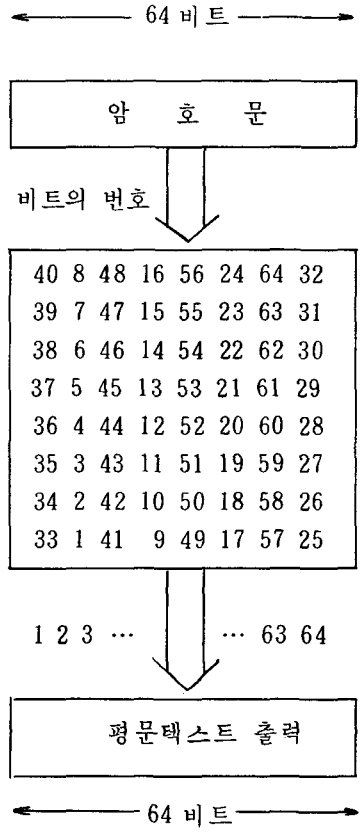


그림 2 - 6 . 逆順列過程

위의 그림에서 보면 初期順列 過程중에 1번 비트가 40번 비트로 위치변환이 됐으므로 逆順列 過程에서 40번 비트를 1번 비트로 위치변환을 시켜준다. 키이에 의한 변환과정은 다음 〈그림 2 - 7 〉과 같다.

解讀過程은 暗號化過程의 逆順으로 進行되며 이때 사용되는 키이는 暗號化過程에서 사용된 키이와 同一한 것을 사용해야 한다.

(3) DES 알고리즘의 하드웨어的인 具現

DES알고리즘은 많은 製作者들이 구조적 특징, 효율성, 제작비용 등을 고려해서 集積回路 (IC : Integrated Circuit)로 具現하여 秘話機를 만드는데 응용되었다. 원래 DES알고리즘은 暗號化와 解讀過程이 64비트 단위로 同時에 처리하도록 되어 있지만 실제 데이터處理에 있어서는 항상 64비트 단위로 처리

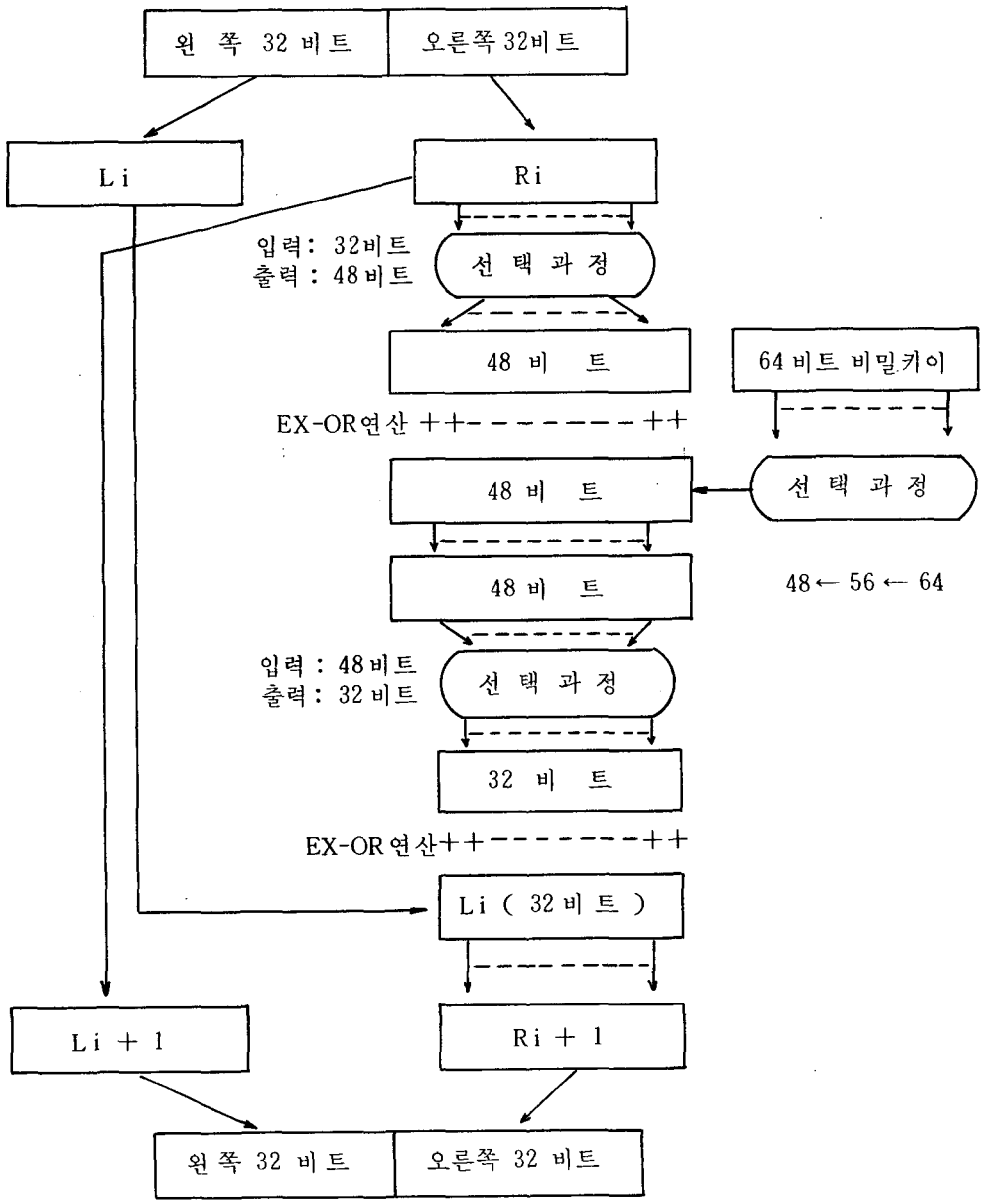


그림 2 - 7 . 키이에 의한 反復處理 변환과정

되는 것은 아니다.

(개 ECB (Electronic Code Book)方法

ECB方法은 DES를 응용한 것 중에서 가장 基本的인 方法으로서 64비트 단위로 暗號化 및 解讀하는 方法이다.

이 方法은 平文 데이터블럭을 DES의 入力 레지스터 (Register) 안으로 받아들여 暗號化해서 出力 레지스터로 내 보내는 方法으로

〈그림 2-8〉과 같다.

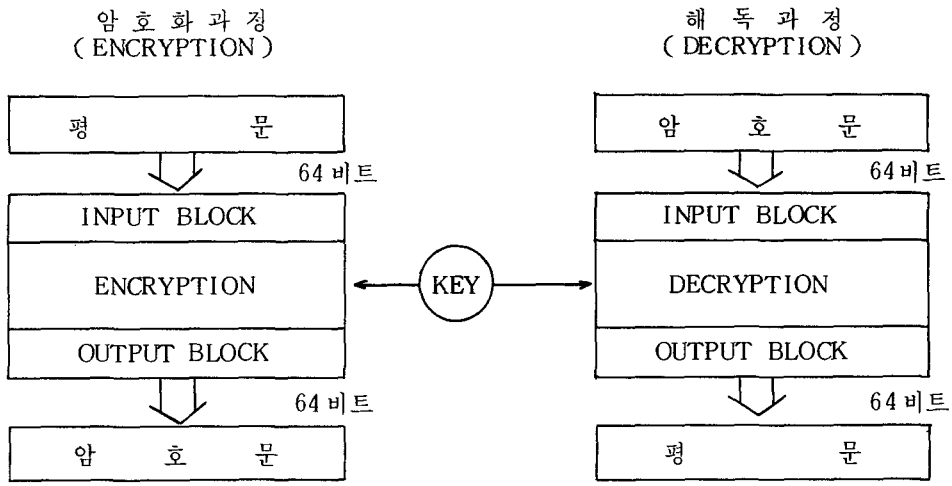


그림 2-8. ECB (Electronic Code Book) 方法

이것은 키이가 일정하면 같은 平文에 대해서 항상 같은 暗號文이 나오기 때문에 키이의 保安이 매우 중요하다.

(나) CBC(Cipher Block Chainning) 方法

CBC方法은 ECB方法의 短點인 블럭의 獨立性(block independence)을 補完하기 위해 chainning機能을 추가시킨 方法이다. 이 方法의 暗號化 및 解讀過程은 〈그림 2-9〉와 같다.

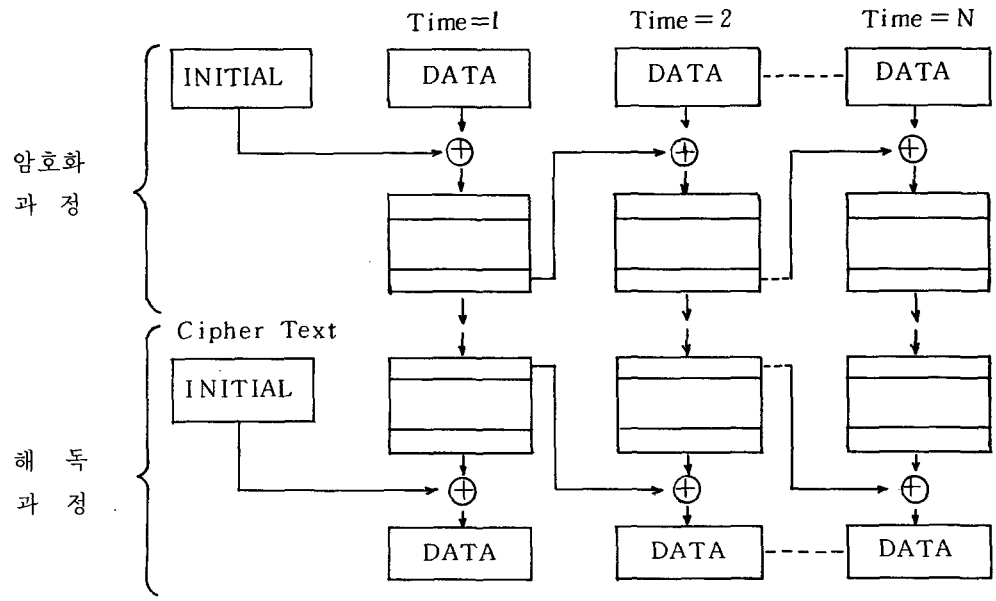


그림 2-9. CBC (Cipher Block Chainning) 方法

暗號化過程은 暗號化해야 할 平文과 前段階에서 生成된 暗號文을 EX-OR연산을 행한 다음 그 結果를 ECB안에서 暗號化하고, 解讀過程은 暗號文을 우선 ECB안에서 解讀한 다음 前段階에서 生成된 暗號文과 EX-OR연산을 행하여 平文을 얻는다. 따라서 이 方法은 블럭을 連結시킴으로써 데이터의 노출에 대한 문제를 상당히 감소시킬 수 있는 方法이다.

(다) Stream Cipher 方法

이 方法은 DES 알고리즘이 固定된 64 비트 블럭단위로 수행되는 短點을 補完하기 위해서 만들어진 것으로 傳送데이터의 보호에 보다적합한 方法이다.

이 方法에서 DES알고리즘은 BSG (Bit Stream Generator)의 일부분으로 사용되며 暗號化過程은 〈그림 2-10〉과 같이 BSG의 出力과 平文이 結合하여 暗號文을 만들고, 解讀過程은 暗號文과 BSG의 出力이 結合하여 원래의 平文을 만들어 낸다.

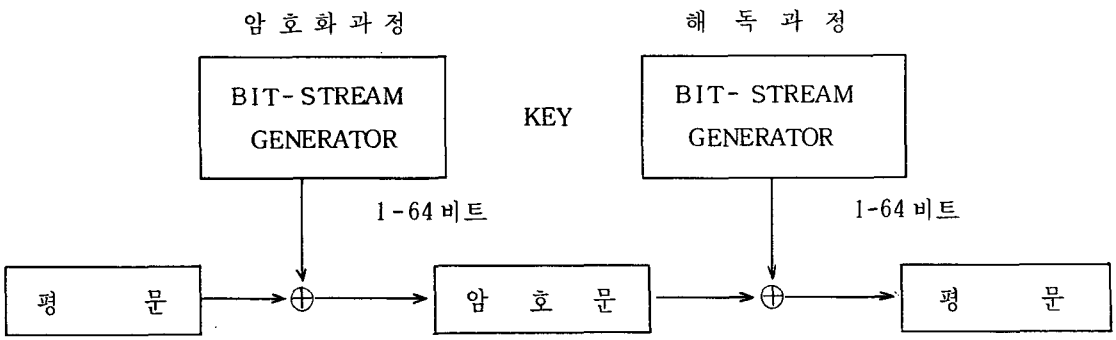


그림 2-10. Stream Cipher 方法

3. 乱數發生 및 키이管理

本 論文에서 사용한 擬似亂數 發生方法은 다음과 같은 要素들을 만족해야 하는데 이 要素들은 擬似亂數가 갖추어야 할 특징이다.

첫째, 發生된 난수와 發生될 난수는 相互獨立이고, 相關關係에 있어서 無關해야 한다.

둘째, 發生된 난수들은 一樣分布 (uniform distribution)를 이루어야 한다.

셋째, 發生되는 난수의 範圍안에서는 같은 난수가 發生되지 않아야 하며 같은 난수가 發生되는 週期는 길수록 좋다.

넷째, 發生된 난수들은 再發生이 可能해야 한다.

다섯째, 난수발생속도가 빨라야 한다.

여섯째, 컴퓨터의 記憶用量은 적게 차지해야 한다.

가. 乱数 發生技法

(1) 中央二乘法 (Middle-Square Methood)

이 方法은 任意로 選定된 數를 제곱한 다음 그 결과치의 中央에 있는 數를 抽出해서 난수를 얻는 方法으로써 代數式은 다음과 같다.

(式) $R_{i+1} = R_i^2$ 의 中央에 位置한 값

이 方法은 난수발생 週期가 짧고 틀에 박힌 경향이 있으며 統計的으로 만족할 만한 亂數

性을 갖지 못하며 난수 발생속도도 느리다.

(2) 修正 中央二乘法 (Modified Middle - Square Method)

이 方法은 任意로 選定한 두數를 곱한 다음 그 數의 中央에 있는 數를 抽出해서 난수를얻고, 다음은 여기에 前段階의 난수를 곱해서 中央에 있는 數를 난수로 택하는 過程을 反復하는 方法이다. 이 方法의 代數式은 다음과 같다.

(式) R_{i+1} = ($R_i \times R_{i-1}$)의 中央에 位置한 값

이 方法은 처음에 부여한 初期値에 의해 그 後의 결과는 자동적으로 決定되기 때문에 確率的인 性質을 갖지 못한다.

(3) 合同法 (Congruential Method)

이 方法은 오늘날까지 알려진 것중에서 가장 널리 이용되고 있는 方法이며 代數式은 다음과 같다.

(式) $R_{i+1} = A \times R_i + C \pmod P \geq \phi$

R_0 : 初期値 $R_0 \geq \phi$

A : Multiplier $A \geq \phi$

C : Increment $C \geq \phi$

P : Modulus

위의 式에서 $C = \phi$인 경우를 乘算式合同法이라 하고 $C \neq \phi$인 경우를 混合式合同法이라 한다. 亂數 數列에서 같은 난수가 發生되는 幅을 週期라 하는데 이 週期가 길수록 亂數로서의 가치가 크다. 이 週期는 multiplier "A"와 초기치 Ro를 어떻게 選定하느냐에 달려 있다. modulus "P"는 난수 發生速度와 관계가 있다.

나. 키이管理 技法

키이管理는 開放된 暗號化시스템에서 가장중요한 부분이다. 이러한 키이管理는 暗號化시스템에 必要한 暗號化키이를 만드는 키이生成過程과 生成된 暗號化키이를 分配하는 過程, 그리고 暗號化시스템 안으로 暗號化키이를 挿入하는 過程으로 이루어진다. 키이는 크게 두가지로 分類할 수 있는데 하나는 다른 키이를 暗號化하는데 사용되는 키이暗號化키이 (key-encrypting-key)로서 최초 暗號化시스템을 設置하는 過程에서 分配되어 오랜 期間동안 사용된다. 다른 하나는 傳送되는 데이터의 暗號化에 사용되는 데이터暗號化키이 (data-encrypting-key) 로서 실제 暗號化시스템이 動作될 때 즉석에서 生成되어 자신이 暗號化해야할 데이터가 있을 때 까지만 存在한다.

(1) 키이의 生成과 分配

키이는 使用可能한 n개의 키이중에서 任意로 선택되어야 한다. 즉 모든 키이가 선택될 수 있는 確率이 同等해야 한다. 또한 예상될 수 있는 키이生成方法은 피해야 한다. 키이를 生成 및 分配하는 方法에는 다음과 같이 두가지 方法이 있다.

(가) Printed Form

이 方法은 가장 基本的인 方法으로 中央에서 키이를 生成하여 비밀리에 文書形態로 分配하는 方法이다. 따라서 키이의 統制가 容易하므로 商業用으로 보다는 軍事用으로 사용하기에 적합하다. 한편, 이 方法은 키이의 統制를 안심하고 맡길 수 있는 管理者가 필요하다는 負擔이 따른다. 그러나 信賴할 수 있는管理者만 있으면 安全하다는 것은 오히려 危險을 줄일 수 있다는 것을 意味하기도 한다. 키이로서 길이가 같은 두개의 비밀숫자가 生成되고 이 숫자들이 EX-OR연산을 행한 結果가 실제 暗號化시스템에 사용되는 키이가 된다. 각각의 비밀숫자들이 서로 다른 管理者에게 분리해서 分配되기 때문에 각 管理者는 오로지 자신에게 分配된 하나의 숫자만을 알고

있을 뿐 상대방의 비밀숫자는 모르게 된다. 각 管理者가 자신의 비밀숫자를 入力하면 두개의 키이가 秘話機안에서 結合하여 하나의 새로운 키이, 즉 兩側이 모두 모르는 키이를 만들어 낸다.

(나) Electronic Form

이 方法은 키이가 하드웨어的으로 키이모듈 (key module) 이나 키이로더(key loader) 안에 電氣的인 形態로 미리 구성되어 있어서 個人的으로 키이를 볼 수 없기 때문에 키이에 대한 秘密을 최대한으로 유지할 수 있다. 또한 키이버튼 (key button) 을 누르면 키이가 自動으로 暗號化시스템에 挿入되기 때문에 키이를 잘못 入力하거나 잘못 읽는 위험이 없다. 그러나 이러한 모듈은 완벽한 保安措置가 취해져야 한다.

(2) 키이의 管理方法

키이管理方法에는 商業的으로 가장 자주 쓰이는 二重키이방법이 있고, 다른 하나는 일명 共用키이 (public key) 라는 것으로서 三重키이방법이 있다.

(가) 二重키이方法

이 方法은 두개의 키이를 사용하는 方法으로서 하나는 秘話機안에서 生成되어 비밀리에 分配하여 다른 키이를 暗號化하는 key-encrypting-key 이고, 다른 하나는 실제 데이터를 暗號化하는데 사용되는 data-encrypting-key 를 사용하는 方法이다.

(나) 三重키이方法

이 方法은 DES 알고리즘과 달리 暗號化와 解讀키이가 서로 다르다. 暗號化키이는 公開的으로 生成하고 解讀키이는 送信側과 受信側이 서로 다른 키이를 비밀리에 유지한다. 이 方法은 일명 共用키이方法이라고 하는데 수행절차를 보면 〈그림 3-1〉 과 같다.

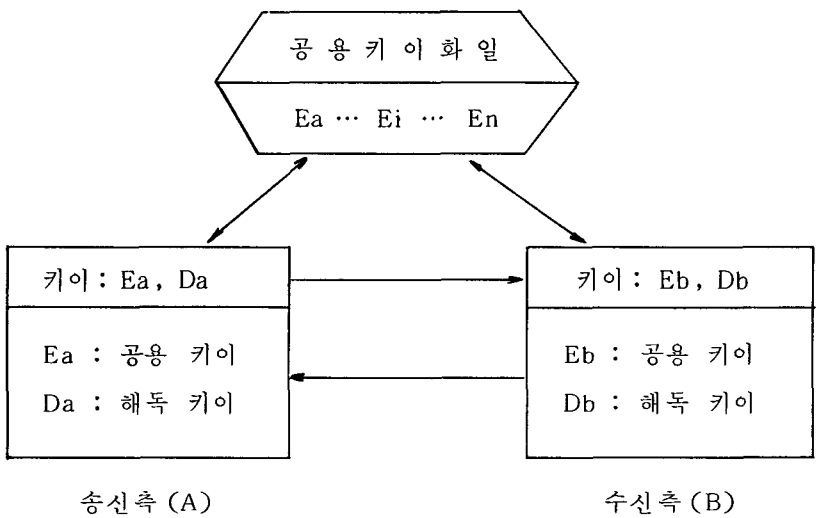


그림 3-1. Public Key System

예를 들어 "A"에서 "B"로 멧세지를 傳送할 경우에 "B"의 共用 暗號化키이(Eb)를 共用키이화일(public key file)에서 얻어 멧세지를 暗號化해서 傳送하면 "B"가 이 暗號文을 受信하여 "B"만이 알고 있는 秘密 解讀키이(D_b)로 解讀한다. 끝으로 키이管理는 시각적으로 보기 좋아야 하고 조작하기 쉬워야 하며 暗號化에 關聯된 모든 사항은 오직 認可者에 대해서만 開放되어야 한다.

4. 暗號化시스템의 設計 및 評價

가. 暗號化시스템의 設計

暗號化시스템은 〈그림 4-1〉과 같이 키이生成過程(KEY-GEN-RTN), 亂數發生過程(RND-GEN-RTN), 테이블運營過程(TBL-GEN-RTN), 暗號化 및 解讀過程(ENC-DEC-RTN)등 모두 4개의 過程으로 구성되어 있다

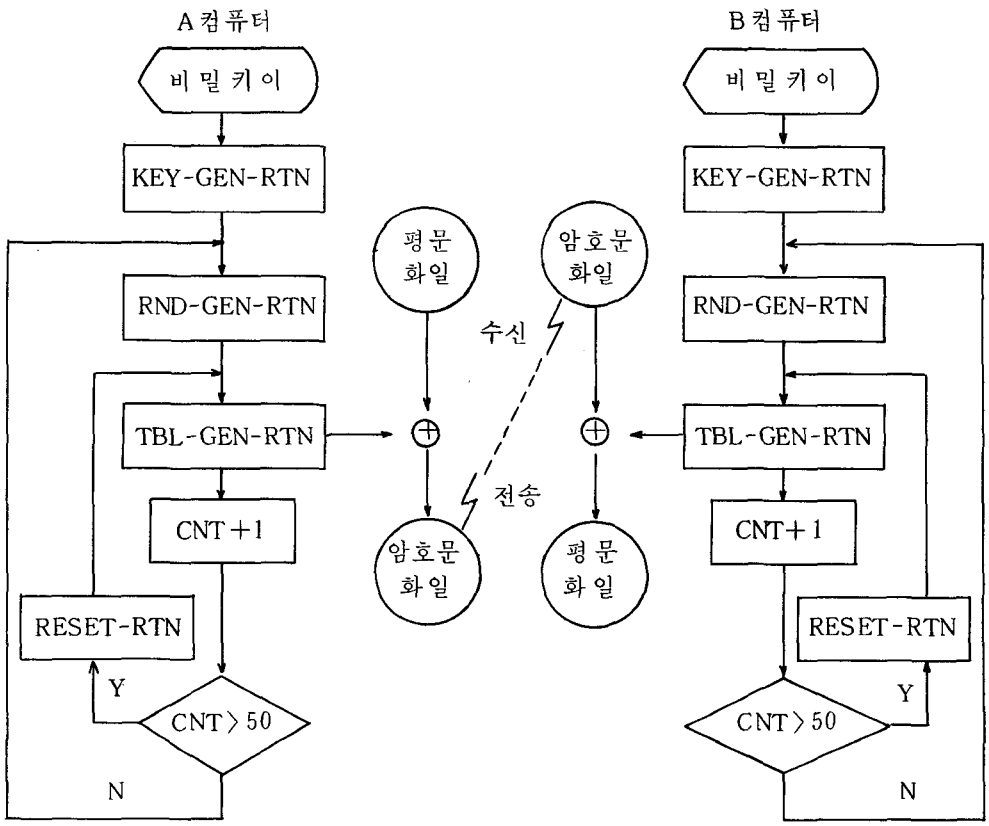


그림 4-1. 暗號化시스템 構成図

暗號化시스템의 수행절차는 다음과 같다. 첫째, 秘密키이가 키이生成過程으로 入力되어 兩側이 모두 예측할 수 없는 새로운 秘密키이가 生成된다. 둘째, 亂數發生過程에서는 새로운 秘密키이를 이용해서 얻은 初期値(key-code)와 곱셈자(seed-value)를 사용하여

亂數를 發生시킨다. 셋째, 테이블運營過程에서는 이 亂數를 받아서 하나의 테이블entry 를 선택하여 出力시킨다. 出力된 文字와 平文의 文字가 EX-OR연산을 수행해서 暗號文字를 生成하여 暗號文화일에 기억시킨다. 넷째, 暗號文字가 生成될 때마다 計數(counter)를 +1씩 증가시켜 計數가 50이 되면, 즉 테이블entry갯수와 같아지면 RESET-RTN이 수행되어 테이블을 再生成하게 된다. 다섯째, 平文이 모두 暗號化되면 暗號文화일을 상대편으로 傳送한다. 解讀過程에서는 暗號文화일의 文字들이 테이블entry와 EX-OR연산을 행해서 平文을 만들어 내는 것을 除外하고 暗號化過程과 同一하다.

(1) 키이生成過程 (KEY-GEN-RTN)

키이生成過程의 基本論理는 다음 〈그림4-2〉와 같다.

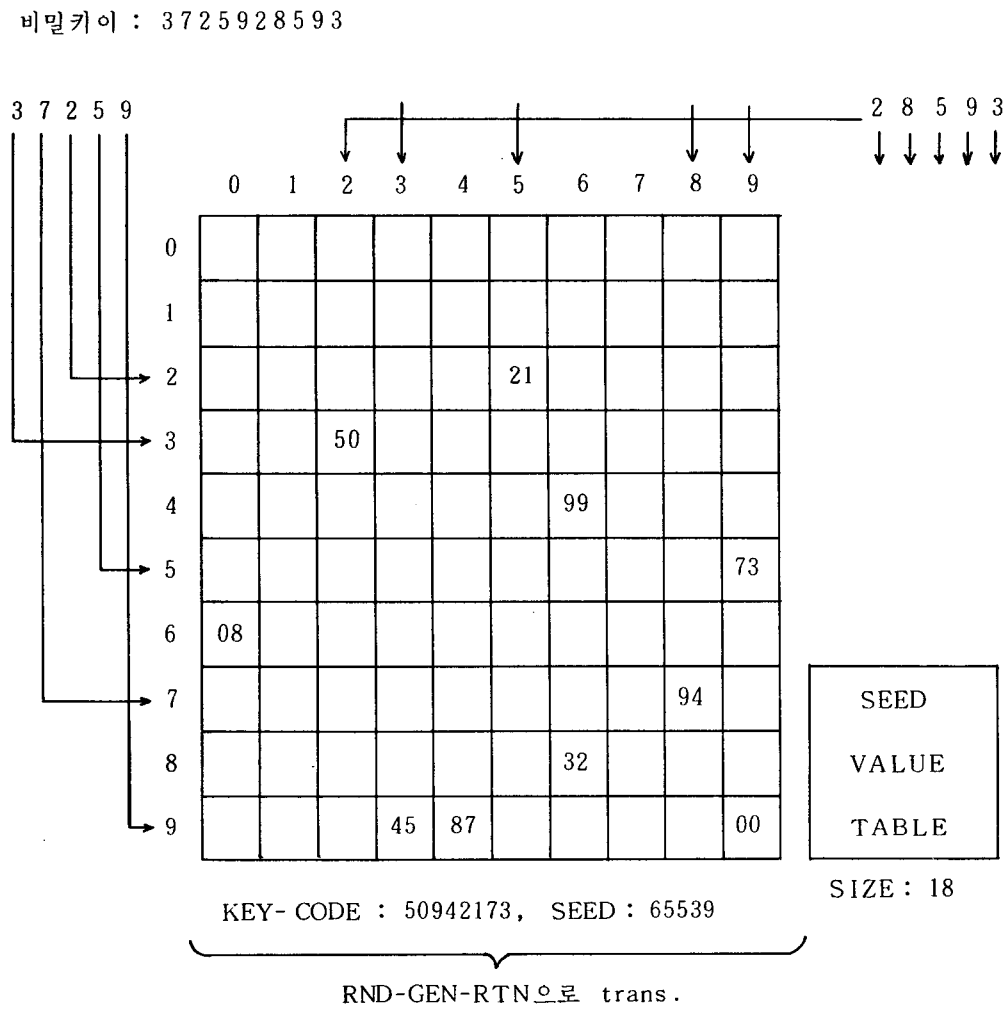


그림 4 - 2 . 키이生成 論理

하나, 테이블의 크기는 10 × 10 으로 정하고 ○○에서 99까지의 두자리 숫자를 任意의 順序로 테이블에 채워 넣는다.

둘, ○○는 테이블의 (1○, 1○)에 位置시키고 秘密키이는 1에서 9^{10} 까지의 열자리 숫자로 制限한다.

셋, 수행절차를 보면 앞의 다섯자리 숫자는 테이블의 row(行) selector가 되고, 뒤의 다섯자리 숫자는 테이블의 column(列) selector가 된다. 예를들어 비밀키이가 "372592 8593"이라고 假定하면 다음과 같은 절차에 의해 수행된다.

〈 KEY - CODE 生成過程 〉

3 row 2 col ⟶ 50 ⟶ 50×10^6 ⟶ TEMP 1에 SAVE

7 row 8 col ⟶ 94 ⟶ 94×10^4 ⟶ TEMP 1에 ADD

2 row 5 col ⟶ 21 ⟶ 21×10^2 ⟶ TEMP 1에 ADD

5 row 9 col ⟶ 73 ⟶ odd, even check

‖

(X라고 하면)

※ 최종적으로 KEY-CODE를 홀수로 만들어 주는 이유는 홀수의 초기치를 사용할 경우에 최대의 난수 발생주기를 갖기 때문이다.

X (mod 2)

= 0 (even) ⟶ +1

≠ 0 (add)

TEMP 1에 ADD

(TEMP 1 = 50942173)

↓

RND-GEN-RTN의 초기치 (X_0)로 사용

〈 SEED- VALUE 生成過程 〉

9 row 3 col　　87　　87 (mod 19) = 11

여기서 "11"은 SEED-VALUE-TABLE의 11번째 entry를 선택하는 인덱스(index)로 사용한다. SEED-VALUE-TABLE은 총18개의 entry를 갖는데 이들은 모두 다섯자리 숫자로서 끝자리는 홀수이고 테이블에 채워넣는 順序는 任意로 정한다. 키이生成過程의 흐름圖(flow chart)는 다음 〈그림 4 - 3 〉과 같다.

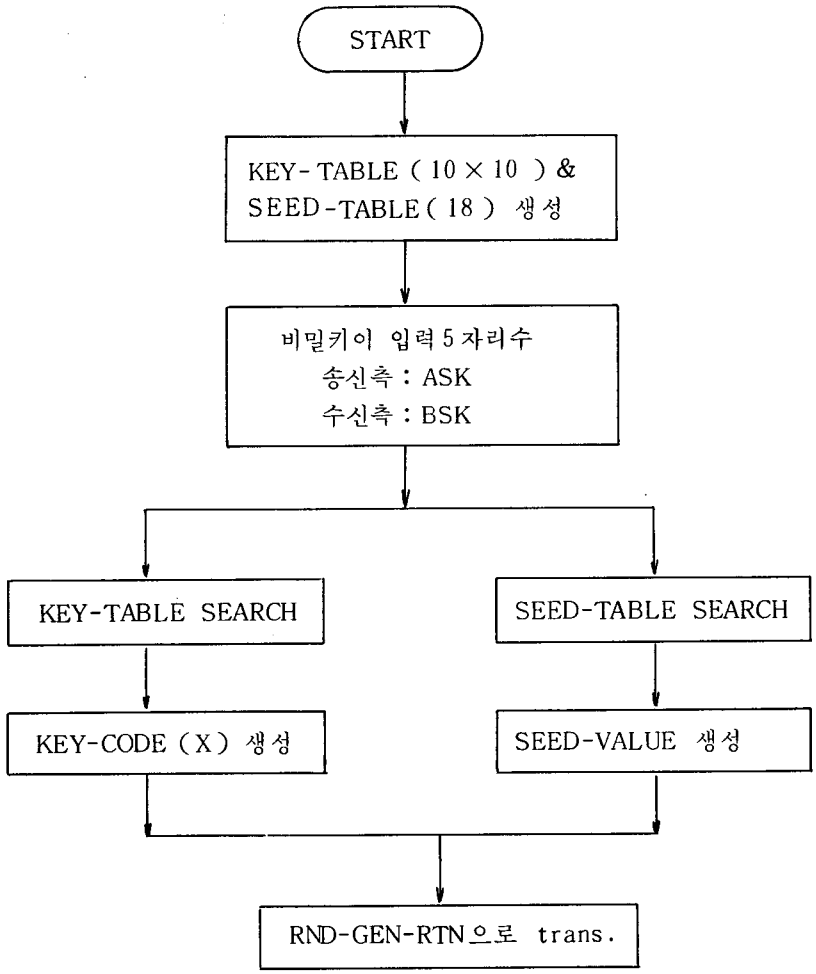


그림 4 - 3 . 키이生成過程 흐름図

(2) 亂數發生過程 (RND-GEN-RTN)

亂數의 發生은 乘算式合同法 (multiplic - ative congruential method)을 사용하여 發生시킨다.

〈公式〉 $X_{n+1} = M \times X_n \pmod P$

여기서 " P "는 modulus로서 2^{31} (214 7483648)을 취한다. 왜냐하면 " P "를 2^{31}로 취할 경우 32비트 CPU를 가진 컴퓨터에서 計算速度가 가장 빠르기 때문이다.

" M "은 seed-value로서 KNUT의 H의 " The art of Computer Programming ", Vol.2의 亂數發生函數 부분중 곱셈자에 관한 定理에서 " P "가 2의 累乘일 때 " M "을 3 or 5 = M (mod 8)인 數를 택하면 가장 많은 亂數가 나타난다는 理論에서 실제로 M = 65539를 택하여 계산해 본 결과 53687 0913번 수행해야 원래의 初期値 (X_0)가 나온다는 것을 알아냈다.

key-code는 乘算式合同法 公式에서 X_n(또는 初期値일 경우 X_0)을 나타내는 데 키이生成過程에서 얻어낸 여덟자리 숫자이다.

seed-value는 3 or 5 = M (mod 8) 인 M으로서 이 條件에 맞는 임의의 다섯자리 숫자 18개를 만들어 테이블로 形成한 것이 SEED-VALUE-TABLE이다. seed-value를 선택하는 方法은 KEY-GEN-RTN에서 나온 seed값에 대해 mod 19를 취해서 이에 해당하는 테이블의 entry를 seed-value로 선택한다. 亂數發生過程의 흐름圖는 다음 〈그림 4-4〉와 같다.

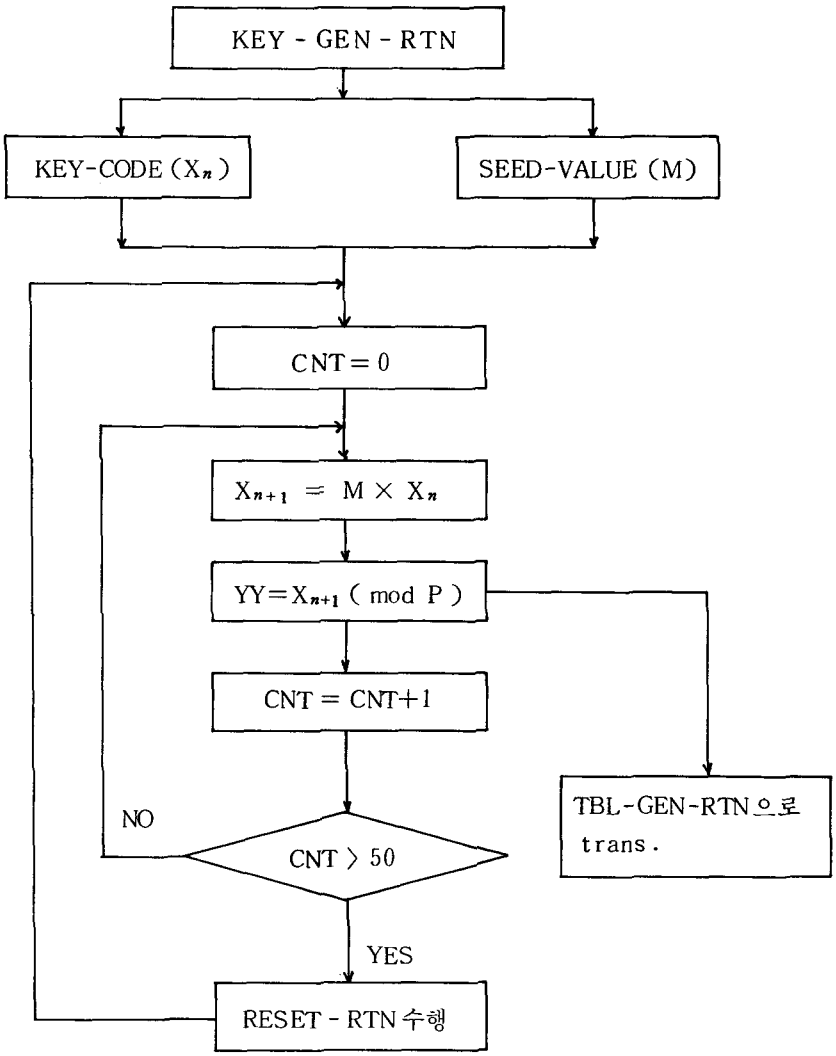


그림 4 - 4. 乱数發生過程 흐름図

(3) 테이블運營過程 (TBL-GEN-RTN)

컴퓨터에서 통상 사용하는 文字중에서 알파벳 大文字 26개 (A～Z), 숫자 (0～9), 特殊文字 14개 (blank 包含)를 취하여 이들을 임의의 順序로 配置하여 테이블을 生成한 다음에 亂數發生過程에서 나온 값 (YY : 인덱스로 사용)에 해당하는 테이블 entry를 선택하여 出力시킨다. 테이블 entry를 한번 出力시킬 때마다 CNT를 +1씩 증가시켜 CNT가 50이 되면 테이블의 인덱스를 변경시켜 새로운 테이블을 生成한다. 새로운 테이블을 生成하는 原理는 처음 生成된 테이블의 인덱스를 K라 하고 다시 같은 크기의 테이블을 잡아 이것의 인덱스를 N이라고 하면 $N = L \times K \pmod{51}$의 公式에 의해 새로운 인덱스 N을 만들어 그 자리에 인덱스 K에 해당하는 테이블 entry를 채워 넣는 것이다. 이 때 선택하는 L은 50以下인 數 중에서 51의 약수인 3과 17의 倍數가 아닌 數, 즉 L은 51의 相對素數이어야 한다.

이 原理에 의해 최초에 만들어진 테이블이 그 다음에 生成된 테이블과 같을 경우는 없으며 그 다음 다음에 生成되는 테이블과 같을確率은 $32/30^{10}$ 이므로 무시할 수 있다. 이 方法으로 만들어낼 수 있는 테이블의 갯수는 약 3×10^{64} (50!)개가 된다. 테이블運營過程의 흐름圖는 다음 〈그림 4-5〉와 같다.

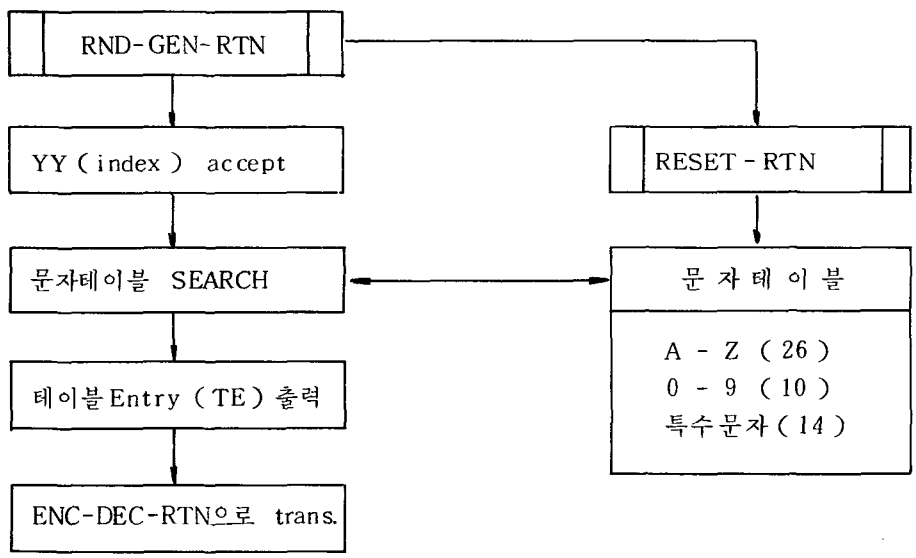


그림 4-5. 테이블運營過程 흐름図

(4) 暗號化 및 解讀過程

暗號化過程은 暗號化할 平文화일에서 文字를 하나씩 읽어 테이블運營過程에서 出力되는 Entry (TE)와 EX-OR연산을 해서 暗號文字를 生成하여 暗號文화일에 기억시킨다. 平文화일의 文字가 모두 暗號化되면 暗號文화일을 受信側으로 傳送한다. 解讀過程은 受信된 暗號文화일에서 文字를 하나씩 읽어 테이블運營過程에서 出力되는 Entry (TE)와 EX-OR 연산을 행하면 원래의 平文으로 환원된다. 暗號化 및

解讀過程의 흐름圖는 다음 〈그림 4 - 6 〉과 같다.

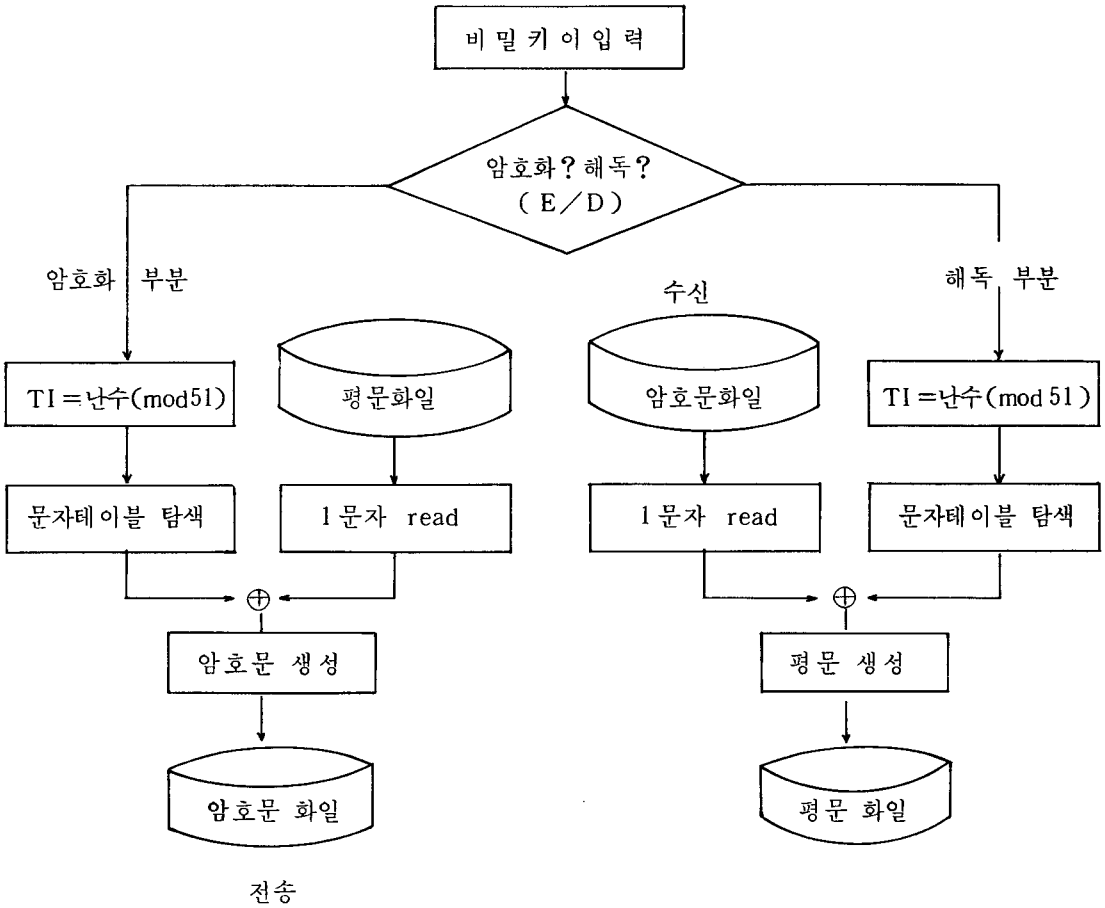


그림 4 - 6 . 暗號化 및 解讀過程 흐름図

나. 暗號化시스템의 評價

本 論文에서 제시한 暗號化시스템을 評價하기 위해 다음과 같은 前提條件을 수립하였다.

하나, 완전한 暗號文이 送受信되는 것으로 假定한다. 즉 送受信할 때나 暗號文 生成시 에러가 發生하지 않는 것으로 假定

둘, 本 論文에서 제시한 暗號化 알고리즘은 公開된 것으로 假定한다. 즉 키이의 길이나 처리절차는 모두 公開되고, 단지 문자테이블의 順序나 새로운 秘密키이 生成테이블의 順序는 使用者가 非週期的으로 變更하여 使用한다.

셋, 不法的 盜聽者는 解讀方法중에서 秘密키이의 反復的 試圖方法을 사용하여 解讀하는 것으로 假定한다.

넷, 不法的 盜聽者는 1 μsec (10^{-6} sec) 당 1개의 秘密키이 테스트가 可能한 매우 처리속도가 빠른 컴퓨터를 사용하여 解讀하는 것으로 假定한다.

위의 前提條件下에서 暗號化시스템을 評價해 보면 다음과 같다.

① 秘密키이는 10자리의 亂數를 使用하므로 총 10^{10}개의 秘密키이를 使用할 수 있다.

② 곱셈자는 5자리의 숫자를 使用하므로 총 10^5개의 곱셈자를 使用할 수 있다.

③ 따라서 ①과 ②에 의해서 使用할 수 있는 키이의 총 갯수는 $10^{10} \times 10^5 = 10^{15}$개가 된다.

④ 秘密키이의 反復的 試圖方法을 使用한 解讀時間을 計算해 보면 다음과 같다.

10^6 (1μsec) × 60 sec × 60 min × 24 hr × 365 day ≒ 3×10^{13} (단위 : year)

$$\frac{\text{비밀키이의 총갯수}}{10^{13}} = \frac{10^{15}}{10^{13}} = 10^2 \text{ year}$$

위의 계산결과에 의하면 不法 解讀時間은 약 100年이 소요된다. 단, 여기에는 文字테이블의 총 경우수인 50 ! = (≒ 10^{64})이 전혀 고려되지 않았다. 그러므로 文字테이블을 수시로 變更시켜 使用하면 解讀하기가 더욱더 어렵게 된다.

⑤ 暗號化시스템의 처리속도를 測定해 본 結果는 다음과 같다. MV-8000Ⅱ (DATA GENERAL社 製作, 32비트 CPU, 主記憶裝置 용량 : 4MB)를 使用하고, 5000개의 文字를 處理하는데 1.97초 (CPU 시간)가 소요되었다. 결과적으로 처리속도나 秘度面에서 볼 때 信賴性이 높은 것으로 評價되었다. 아울러 文字테이블을 不規則的으로 變更시켜서 運營하면 더욱더 效果的인 暗號化시스템이 될 것이다.

5. 結 論

위에서 살펴 본 바와 같이 暗號化 시스템에서 가장 중요한 키이의 管理에 있어서는 최대한의 保安을 유지할 수 있도록 최초 分配된 秘密키이를 入力하면 이것에 의해 새로운 秘密키이가 生成됨으로써 키이生成테이블의 順列을 모르고는 거의 解讀할 수 없도록 하였다.

또한 二重 保安方策으로써 亂數發生에 必要한 곱셈자를 여러개 生成하여 이중에서 하나를 임의로 선택할 수 있도록 하였다. 그러나 暗號化시스템이 노출되었을 경우를 對備하여 非週期的으로 곱셈자들을 變更시켜 주거나 키이生成테이블과 文字테이블의 entry 位置를 變更시켜 주는 것이 바람직하다. 本 論文에서 提案된 暗號化시스템이 效果的이긴 하지만 알고리즘이 公開되었을 경우에 時間은 많이 소요 되겠지만 解讀될 可能性이 있다. 따라서 暗號化시스템에서 가장 핵심이 되는 키이生成 部分을 micro-chip으로 제작하여 物理的인 保安裝置를 추가하여 철저하게 保安을 유지하고, 秘話通信을 수행하기 前에 送信側과 受信側이 相互 상대방을 確認할 수 있는 暗號化시스템이 추가된다면 더욱 더 安全하고 處理速度도 向上된 暗號化시스템이 될 수 있을 것이다. 아울러 모든 경우의 可能性에 對備한 完全한 시스템을 開發하는 데 앞으로 繼續的인 硏究가 必要하다.

參 考 文 献

1. Denning, Cryptography and Data Security, Addison-Wesley Co., 1983.
2. Tanenbaum, Computer Networks, Prentice-Hall Inc., 1981.
3. Knuth, Seminumerical Algorithms: The Art of Computer Programming Vol. 2, Addison-Wesley Co., 1973.
4. Donn B. Parker, Computer Security Management, Reston Publishing Co., 1981.
5. L.J. Hoffman, Security and Privace in Computer Systems, Melville Publishing Co., 1973.
6. William Stallings, Local Networks: an Introduction, Macmillian Publishing Co., 1984.
7. Martin E. Hellman, A Cryptanalytic Time-Memory Trade-off, IEEE Trans. Information Vol. LT-24, 1980.
8. Abbruscato R.C., Data Encryption Equipment, IEEE Communication magazine, Vol. 22, No. 9, September 1984.
9. Stephen T. Kent, Network Security: a Top-down View Shows Problem, MIT Laboratory, Cambridge, Mass., 1983.
10. James Martin, Security Accuracy and Privacy in Computer Systems, Prentice-Hall INC., 1973.
11. Alam G. Konheim, Cryptography: A Prime, John-Wiley & Sons Inc., 1981.
12. Carl H. Meyer, Cryptography: A New Dimension in Computer Data Security, John-Willey & Sons Inc., 1982.
13. 정진욱, 변옥환, 데이터통신과 컴퓨터 네트워크, Ohm社, 서울, 1982.
14. 이동한, 양해술, EDP 시스템 시뮬레이션, 일조각, 서울, 1982.