通信情報保護學會誌
第2卷 第3號, 1992. 9

난수와 암호

양형규* · 안영화**

1. 서 론

현대사회가 정보화 사회로 발전해감에 따라 컴퓨터 통신망상의 데이타 트렌젝션이 증가하게 되었고, 이에 따라 전송중인 데이타에 대해 보안대책이 필요하게 되었다. 보안대책에는 여러가지 방식이 있으나, 데이타의 분실이나 도난 혹은 도청으로 부터 데이타를 보호할 수 있는 가장 효과적인 방법으로 암호화 방식이 이용되고 있다.

암호화 방식에는 DES같은 관용키 시스템, [DH]이 제안한 공개키 시스템, 그리고 영지식 상호 증명방식[GMR]을 이용한 Id-base 시스템 등이 있으며, 이러한 암호화 방식은 안전성을 제공하고 비밀키를 생성하기 위한 방법으로 난수(혹은 의사난수)를 사용하고 있다. 난수는 이 외에도 Monte Carlo 방식(e.g, Monte Carlo primality test)과 같은 응용분야 및 효율적인 확률적 알고리즘에서 유용하게 사용되고 있다.[M][GM]

"Encyclopedic dictionary of mathmathics"에 의하면 난수는 일련의 랜덤 숫자들(a sequence of random numbers)이 독립적이며 이상적으로 분포되어 있는 랜덤 변수들(random variables)의 실체들로서 간주할 수 있는 일련의 숫자들이다. 암호학에서는 완전한 랜덤 수열(truly random sequence) 대신에 의사난수, 즉 효율적이고 결정적 알고리즘(deterministic algorithm)에 의해서 생성된 확실히 랜덤한 것처럼 보이는 유한한 수열들을 사용하고 있으며 이러한 수열을 발생하는 알고리즘을 의사난수 생성기라 한다.

이와같은 난수는 암호학에서 폭넓게 사용되고 있는 중요한 도구이므로, 암호학 관점에서의 난수(혹은 의사난수)에 대한 정확하고 실질적인 정의가 필요하며, 완전한 랜덤 수열과 유사한 난수를 발생하는 의사난수 생성기에 대한 정의 및 구성방법 그리고 이러한 난수(혹은 의사난수)와 암호와의 관계(즉, 암호방식에서 사용되는 난수의 용법)에 대한 고찰이 필요하게 되었다.

따라서 본고에서는 난수 및 의사난수에 대해 암호학적 관점에서 정의하고, 이러한 의사난수를 생성하는 의사난수 생성기의 종류 및 구성방법을 설명하고자 한다. 또한 의사난수 생성기에 의해서 생성된 의사난수가 크립토시스템의 어떤 분야에서 사용되고 있는지 그리고 어떻게 적용되고 있는지 예를 들어 고찰하고자 한다.

2. 난 수

2.1 정 의

난수란 확률변수열이 가지는 성질(서로 독립이며

* 성균관 대학교 정보공학과
** 강남대학교 전자계산학과

모두 일양 분포를 따름)을 모두 만족하는 수이다. 이와같은 완전한 난수를 발생시키는 것은 실질적으로 불가능하므로 암호방식에서는 의사난수를 사용하고 있다. 의사난수란 다항식 시간내에 서로 독립이고 일양분포로 나온 수(정상적인 동전을 던져서 만든 같은 길이의 수)와 구별할 수 없는 것이다. [BM]

의사난수 생성기란 모든 실제적 사용목적에 있어서 완전한 랜덤 수열과 유사한 출력비트열, 즉 의사난수를 생성하기 위해서 사용되는 알고리즘이라고 정의할 수 있으며, 또한 랜덤 seed를 입력 받았을 때 매우 긴 의사난수 비트열을 출력하는 결정적 알고리즘이라고 다시 정의할 수 있다.[Go] 의사난수 생성기를 수식적으로 정의하기 위해서 사용되는 개념은 크게 두가지로 구분하여 구별 불가능(indistingushable) 개념과, 예측 불가능 분포(unpredictable distribution)개념이다.

먼저 구별 불가능 개념을 사용해서 정의해 보자. 여기서 U_n은 길이 n인 스트링들이 균일하게 분포되어 있는 랜덤변수(random variable)라 하자.

정의 1 : 만약 X_n이 U_n과 구별 불가능 하다면 일련의 랜덤변수들 $\{X_n\}_n$은 의사난수(pseudorandom)라 한다. 단, 모든 n에 대해서 X_n은 $\{0, 1\}^n$으로 구성된 스트링들 상의 랜덤변수들이다.

정의 2 : X_n이 다음과 같은 조건을 만족하면 polynomially sampled 랜덤변수라 한다.

조건 : $Prob[S(1^n, r)=\alpha]=Prob[X_n=\alpha]$인 다항식시간 샘플링 알고리즘이 존재

단, 확률은 uniform coin tosses r과 관련

정의 3 : 결정적 다항식 알고리즘(deterministic polynomial algorithm) G(r)이 다음의 두 조건을 만족하면 의사난수 생성기라 한다.

조건 1 : $\forall r \quad |G(r)| > |r|$

조건 2 : 랜덤변수 비트열(random variable sequence) $G(U_n)$은 의사난수

정의 3에 대해서 고찰하면 우선 G는 스크래치(scratch : 0으로 구성된 스트링)로 부터 랜덤샘플(random samples)을 생성할 수 없다. 이것은 G가 결정적 알고리즘이기 때문에 불가능하다. 오히려 G는 랜덤입력 seed를 랜덤특성들(random properties)을 유지시키면서 긴 스트링으로 확장시키기 위해서 사용된다. 그리고 조건 1은 G의 출력이 자신의 입력보다 더 길다라는 것을 보장한다. 또한 정의 3은 같은 크기인 두개의 입력들에 대해서 다른 출력크기(output lengths)을 생성하는 의사난수 생성기를 허용한다. 본고에서는 입력의 길이가 동일하면 출력의 길이도 동일한 의사난수 생성기만 고려할 것이다.

다음은 예측 불가능 분포 개념을 사용하여 의사난수 생성기를 정의해 보자. 완전한 랜덤이진 수열들의 주목할 만한 특징은 스트링의 prefix을 안다고 해서 다음 비트에 대한 어떠한 정보도 얻을 수 없다는 점이다. 따라서 본고에서는 pseudorandom distribution에 대응하는 특성을 먼저 정의하고 이에 따른 의사난수를 정의하고자 한다.

정의 4 : 만약 확률분포에 따라서 선택된 스트링들의 prefix를 입력으로 하는 모든 확률적 다항식 시간 알고리즘이 negligible advantage을 가지고 스트링의 다음 비트를 예측하는데 성공하면 이 확률분포는 the next bit test를 통과했다고 한다. 즉, $\{X_n\}_n$은 다음의 조건을 만족하면 the next bit test를 통과한다.

조건 : 모든 확률적 다항식 시간 알고리즘 A와 임의의 상수 c에 대해서 $|Prob[A(w)=\sigma]-1/2| < 1/n^c$을 만족

단, 확률은 X_n에서 선택된 스트링들의 prefix인 $w \circ \sigma$ 스트링들과 관련

정의 4에 의해서 the next bit test를 통과하는

분포는 예측 불가능하다고 한다. 이상과 같은 개념을 사용해서 의사난수를 다음과 같이 정의할 수 있다.

정의 5 : 만약 polynomially sampled distribution이 the next bit test를 통과하면 그 수열은 의사난수이다.

2.2 난수 생성기의 존재

의사난수 생성기는 크립토시스템의 여러 분야에서 많이 적용되고 있으며, 그 역할 또한 중대하다고 할 수 있다. 그 예로서 관용키 시스템에서는 관용키를 의사난수 생성기의 seed로 사용하고, 공개키 시스템에서는 비밀키를 생성하기 위해서 의사난수 생성기를 사용하였다. Bit-commitment 방식에서도 의사난수 생성기를 사용하며, 영지식 상호증명 방식에서는 의사난수 생성기와 난수를 이용하고 있다. [FFS] 특히 multi-pahty 프로토콜에서는 각 참가자의 비밀 입력에 대한 정보를 감추기 위해서 의사난수 생성기를 사용한다[BG]

따라서 이와같은 의사난수 생성기를 구성할 수 있는가 혹은 의사난수 생성기가 존재하는가에 대해서 간략히 알아 보고자 한다.

여러 논문에서 의사난수 생성기의 존재에 대해서 논하였으나, 본고에서는 Goldreich가 제안한 이론을 설명하고자 한다. Goldreich는 one-way permutation이 존재하면 의사난수 생성기가 존재한다고 증명하였다. 이것은 모든 다항식 Q(n)에 대해서 길이 n인 스트링들을 길이 Q(n)인 스트링들로 출력하는 의사난수 생성기가 존재한다고 말할 수 있다. 또한 one-way permutation이 존재하면, 그때 자신의 입력을 한 비트씩 확장할 수 있는 의사난수 생성기가 존재한다라고 하였으며, 자신의 입력을 한 비트씩 확장할 수 있는 의사난수 생성기가 존재하면, 그때 모든 다항식 Q(n)에 대해서 길이 n인 모든 입력을 길이 Q(n)인 스트링으로 확장시키는 의사난수 생성기가 존재한다라고 하였다. 이 외에도 Noar는 one-way function이 존재한다면 의사난수 생성기가 존재한다고 증명하였으며[No], [Y]논문에서 의사난수 생성기의 존재는 BPP $\subset$ DTime(2^{n^k}) (단 $\forall k > 0$)을 의미한다라고 증명하였다. 이상과 같이 의사난수 생성기는 계산복잡도 관점에서 볼 때 존재할 수 있음이 증명되고 있다.

3. 이진 난수 발생기(Random bit Generator)

이진 난수 발생기는 사용하는 키 스트림 형태에 따라 주기성과 비주기성으로 나누어 진다. Rotor Machine이나 Hagelin Machine으로 만든 이진 난수는 주기적이며, Vernam이나 Running Key는 비주기적이다. 또한 이진 난수 발생기는 동기식(Synchronous)과 자동 동기식(Self-synchronous) 방식으로 나눌 수 있다.

동기식 이진 난수 발생기는 키 스트림이 평문이나 암호문의 스트림과 독립으로 생성되고, 그 종류로는 선형귀환 쉬프트 레지스터와 비선형귀환 레지스터 방식이 있으며[Choi], 자동동기식 이진 난수 발생기는 키 스트림이 앞의 암호문 스트림에 의존하여 생성되며, 종류로는 Antokey 방식과 CFB(cipher feedback)방식이 있다.

3.1 선형귀환 쉬프트 레지스터 (Linear feedback shift register)

쉬프트 레지스터에 의해 생성되는 이진 수열은 $b_i=(a_1b_{i-1} + \cdots + a_db_{i-d}) \bmod 2$에 의해서 얻어진다. 위의 식은 다항식 $f(x)=x^d+a_1x^{d-1} + \cdots + a_d$에 대응된다. 여기에서는 $f(x)=x^p+x^q+1$과 같은 형태의 다항식을 사용한다.

위의 특성 다항식에 대응하는 쉬프트 레지스터를 이용하여 이진수열을 생성하는 알고리즘은 다음과 같다.

LFSR 알고리즘

단계 1) $Y \leftarrow X$(X는 b_{i+p-1}, $b_{i+p-2} \cdots b_i$의 형태로 되어 있다.)

단계 2) Y를 q 비트 만큼 오른쪽으로 이동시키고 빈자리는 0으로 채운다.

단계 3) Y ← X ← Y XOR X(여기서 XOR은 exclusive OR 연산을 의미한다.)

단계 4) Y를 p-q 비트 만큼 왼쪽으로 이동시키고 빈자리는 0으로 채운다.

단계 5) X ← Y XOR X(X는 다시 b_{i+2p-1}, b_{i+2p-2} … b_{i+p}로 구성된다.)

쉬프트 레지스터에 의해 생성되는 비트열의 주기는 2^p-1이하이다. 이진 수열 $\{b_i\}$의 주기가 최대주기인 2^p-1이 되는 선형 쉬프트 레지스터를, 최대주기를 갖는 선형 쉬프트 레지스터라 하고 m-LFSR(maximum length Linear Feedback Shift Register)라고 정의한다.

3.2 승산(multiplication)시스템

m-LFSR 2개의 출력을 서로 곱하여 최종의 출력 수열을 발생하는 시스템을 승산시스템이라고 한다. m-LFSR1의 출력수열이 $\{a_i\}$이고 m-LFSR2의 출력 수열이 $\{b_i\}$인 경우 승산시스템의 출력수열 $\{c_i\}$은 c_i은 $c_i=a_i\times b_i$가 된다.

3.3 Geffe 시스템

Geffe 시스템은 3개의 m-LFSR로 구성된다. m-LFSR1, m-LFSR2, m-LFSR3의 출력 수열을 각각 $\{a_i\}$, $\{b_i\}$, $\{c_i\}$라고 하면 Geffe 시스템의 출력수열 $\{g_i\}$는 다음과 같이 생성된다.

$$\{g_i\}=a_ib_i \oplus c_ib_i \oplus c_i$$

m-LFSR1, m-LFSR2, m-LFSR3의 차수가 m, n, k이고 각 쌍마다 서로소가 될 때 Geffe 시스템에서 발생되는 출력수열의 주기는 $(2^m-1)(2^n-1)(2^k-1)$이 된다.

3.4 CFB 방식

Feedback 레지스터 R은 쉬프트 레지스터로서 구성되고, 처음에 I_0로 초기화한다. 암호문 문자 C_i를 생성하면 곧 레지스터 R의 좌단에 shift-in하여 우단의 1문자를 shift-out하여 버린다. 이 R_0의 값을 입력하여 E_k(암호 알고리즘)을 계산하고, 출력 결과 중 최하위 문자를 다음 단계의 키 문자로 한다.

CFB 방식은 1회에 1개의 문자를 조작하고 문자의 길이 m을 설계의 파라메터로 선택할 수 있고, 이것은 m비트 CFB 방식이라 한다. 스트링의 길이는 최소 1비트의 경우 1비트 CFB모드가 되며, 일반적으로 m의 값은 64비트까지 사용할 수 있다. 보통 통신 시스템에서 사용하는 문자의 크기에 의해 m을 선택한다.

4. 의사난수 생성기(Pseudorandom Generator)

의사난수는 완전하게 랜덤한 수를 생성할 수는 없다. 의사난수 수열이 주어졌을때 exhaustive 탐색 방법은 의사난수 수열이 생성되는 seed를 발견할 수 있다. 이 방법은 다항식 시간내에 seed를 계산할 수 있어야만 이용 가능하다.

이러한 의미에서 볼 때 앞으로 설명할 선형합동 생성기(linear congruence generator)와 (1/p)-생성기는 예측 가능하다. 즉, 생성기에 의해서 발생된 충분히 긴 수열을 입력으로 할때 이러한 수열을 생성하는 seed를 출력하는 다항식시간 알고리즘이 존재한다.

그리고 의사난수 수열의 랜덤성을 측정하는 방법으로 "다항식 시간 unpredictability"를 사용할 수 있다. 특히, 앞으로 설명할 평방잉여 생성기(quadratic residue generator)와 지수 생성기(index generator)는 계산복잡도 관점에서 볼 때 그러한 문제(소인수 분해 문제, 이산대수 문제)들은 다항식 시간 내에 풀 수 없다는 가정하에 다항식 시간 예측 불가능하다. 이러한 예측 불가능한 수열들은 모든 다항식 크기의 통계적인 검사(statistical test)를 통과한다. 즉, 다항식 시간 예측 불가능 수열은 임의의 다항식 시간 통계적 검사에 의해서는 같은 크기의 완전한 랜덤 스트링과 구별 불가능하다.

4.1 선형합동 생성기

정수 a, b, m(단, m > max(a, b))은 비공개되고 고정된 양의 정수이고 각각의 정수 x < m에 대해서 무한수열들(x_0, x_1, ···, x_i, ···, x'_1, x'_2, ···, x'_i, ···,)을 다음과 같이 정의한다.

$$x_i \equiv \begin{cases} x & \text{if } i=0 \\ (a \cdot x_{i-1}+b) \bmod m & \text{if } i>0 \end{cases} \quad (1)$$

$x'_{i+1}=(x_{i+1}-x_i)$, 단 $i \geq 0$
그리고 모든 $i \geq 1$에 대해서
$x'_{i+1} \equiv a \cdot x'_i \bmod m$이다. (2)

이때 LGEN(선형합동 생성기)은 위에서 정의된 정수들 〈x, a, b, m〉을 입력으로 받아들이고 출력 LEGN(x, a, b, m)은 무한수열 (x_0, x_1, ···, x_i, ···,)이 된다.

예) LEGN(3, 7, 5, 12)=3, 2, 7, 6, 11, 10, 3, 2, 7, 6, 11, 10, ···

따라서 입력으로 수열 x_0, x_1, ···, x_{t+1} (단, LEGN(x, a, b, m)의 출력)이 주어지고, 다음과 같은 정수들 a', b'을 출력하는 효율적인 알고리즘 A가 존재한다.

단, t= the least ≥ 1 such that $g_i \mid x'_{i+1}$

$$x_i \equiv (a' \cdot x_{i-1}+b') \bmod m \quad (3)$$

이 알고리즘 A는 $\log_2 m$인 다항식 시간내에서 동작한다.

4.2 (1/p)-생성기

p는 odd prime, g는 Z_p^* 상의 원시원소이고 |p|는 base g에서 p의 길이, 즉 $|p| = |\log_g p|$ 이다. $0 \leq r < g$인 정수 r을 g-digit라 하고, $0 < r < g$인 임의의 정수 r이 주어졌을 때 무한수열 r_0, r_1, ···, r_m, ···을 다음과 같이 정의하자.

$$r_m \equiv r \cdot g^m \bmod p, \quad m > 0 \quad (4)$$

g는 Z_p^* 상의 원시원소이기 때문에 Euler-Fermat 정리로 부터 임의의 $0 < r < g$에 대해서 수열 r_0, r_1, ···, r_m, ···의 주기는 p-1과 같다. 그러므로 $\{r_0, r_1, \cdots, r_{p-2}\}=\{1, 2, \cdots, p-1\}$이다.

Euclidean 알고리즘과 r_m의 정의로 부터 다음과 같은 식을 얻을 수 있다.

$$r_i/p=q_{i+1}\ q_{i+2} \cdots\ q_{i+m}+1/g^m \cdot r_{i+m}/p \quad (5)$$
$$r_i g^m=(q_{i+1}\ q_{i+2} \cdots\ q_{i+m})p+r_{i+m} \quad (6)$$

g-digit인 무한수열 $x_1\ x_2$ ···, x_m, ···이 다음의 세 조건을 만족하면 주기는 p-1이고, base가 g인 de Bruijn sequence가 된다.

조건 1) g-digit인 수열 $x_1\ x_2$ ···, x_m, ···은 p-1의 주기를 갖는다.

조건 2) 길이 |p|-인 g-digit의 모든 유한수열은 $x_1\ x_2$ ···, x_m, ···의 세그먼트로써 최소한 한번 발생한다.

조건 3) 길이 |p| 인 g-digit의 모든 유한수열은 $x_1\ x_2$ ···, x_m, ···의 세그먼트로써 최대한 한번 발생한다.

예1) p=3(=11 in base 2)일때, 수열 0, 1, 0, 1, ···은 주기 2와 base 2인 de Bruijn sequence이다.

예2) p=5(=101 in base 2)일때, 수열 0, 1, 1, 0, 0, 1, 1, 0, ···은 주기 4와 base 2인 de Bruijn sequence이다.

PGEN((1/p)-생성기)은 입력으로 〈p, r, g〉을 받아들이고, 출력 PGEN(p, r, g)은 rational number r/p가 base g에 표현될 때 발생되는 g-digits인 무한수열 $q_1\ q_2$ ···, q_m, ···이다.

그리고 mod p의 임의의 원시원소 g와 임의의 g-digit r에 대해서, 수열 PGEN(p, r, g)은 주기 p-1과 base g인 de Bruijn sequence이다.

4.3 평방잉여 생성기(Quadratic Residue Generator)

다음은 충분히 큰 합성수에 대한 소인수 분해의 어려움(소인수 분해 문제는 NP문제에 속함)을 근거로 한 의사난수 생성기를 구성하고자 한다. 임의의 수가 평방잉여인지 평방 비잉여인지를 결정하는 문제의 어려움은 소인수 분해 문제로 귀착된다.[L]

n은 두개의 서로 다른 소수, 즉 $p \equiv q \equiv 3 \bmod 4$의 곱인 정수상에 분포하고, $N=\{N_k : k \in I\}$는 양의 정수들인 nonempty 집합 N_k를 의미한다.(단, I는 무한한 지수들의 집합) 그리고 모든 $n \in N_k$에 대해서 정수 n의 길이는 k이다. squaring mapping $x \rightarrow x^2 \bmod n$ ($x \in QR_n$, $x^2 \bmod n \in QR_n$)은 Jacobi symbol값인 1과 -1을 갖고 그것은 $x \rightarrow \sqrt{x \bmod n}$로 표기되는 역의 근을 갖는다.($x \in QR_n, \sqrt{x \bmod n} \in QR_n$) 여기서 합성수 n을 구성하는 소인수들을 모르면 역의 근을 찾는 문제 또한 NP문제이다. 합성수 N에 대한 임의의 수가 평방잉여인지 평방 비잉여인지를 결정하는 문제의 어려움을 표현해 보기로 하자. 본 절에서 사용할 함수 P, Q는 다항식 시간 함수이며, parity 함수는 다음과 같다.

$$par(x)=\begin{cases} 0 & \text{if } x \text{ is even} \\ 0 & \text{if } x \text{ is odd} \end{cases} \quad (7)$$

우선 parity function에 대한 특성 및 평방잉여를 결정하는 함수의 특성을 정의하면 다음과 같다.[GM]

정의 6 : 다항식 크기 회로(polynomial size circuit) $C=\{C_k : k \geq 1\}$는 다음과 같은 조건을 만족하면 family N에 대한 parity function을 계산하는데 1/P-advantage를 갖는다.(APAR(C, N, 1/2 + 1/P))

조건) $k \in I$인 유한 지수들을 제외한 모든 지수들에 대해서 다음의 특성을 갖는다.

$Pr[x \in QR_n : par(\sqrt{x \bmod n})] \geq 1/2+1/p(k)$. 단, $\forall n \in N_k$

정의 7 : 다항식 크기 회로 $C=\{C_k : k \geq 1\}$는 다음과 같은 조건을 만족하면 family N에 대한 평방잉여를 결정하는데 1/P-advantage를 갖는다.

(AQR(C, N, 1/2+1/P))

조건) $k \in I$인 유한 지수들을 제외한 모든 지수들에 대해서 다음의 특성을 갖는다.

$1/2(Pr[C_k(n, x)=1 \mid x \in QR_n]+Pr[C_k(n, x)=0 \mid x \neq QR_n] \geq 1/2+1/P(k)$

단, $\forall n \in N_k$에 대해서 x는 Z_n^* (+1)

위의 정의에 의해서 평방잉여 생성기를 구성해 보자.

QRGEN(평방잉여 생성기)은 입력으로 한 쌍 $\langle x, n \rangle$을 받아 들이고(단, $x \in QR_n$), 이것의 출력은 …, $b_{n, i-1}(x)$, $b_{n, i}(x)$, $b_{n, i+1}(x)$, …인 무한 수열의 비트들이다.

이러한 무한 수열 …, $b_{n, i-1}(x)$, $b_{n, i}(x)$, $b_{n, i+1}(x)$, …, 은 다음처럼 생성할 수 있다. 즉, 정수 n이 주어질 때 함수를 정의하면 $f_n : QR_n \rightarrow QR_n : x \rightarrow f_n(x)=x^2 \bmod n$이고, 이것의 역함수는 $f_n^{-1} : QR_n \rightarrow QR_n : x \rightarrow f_n^{-1}(x)=\sqrt{x \bmod n}$ 이다.

그리고 함수 f_n^i 는 다음과 같이 정의되고,

$$f_n^i(x)=\begin{cases} x & \text{if } i=0 \\ f_n(f_n^{i-1}(x)) & \text{if } i>0 \\ f_n^{-1}(f_n^{i+1}(x)) & \text{if } i<0 \end{cases} \quad (8)$$

n과 $x \in QR_n$에 대한 비트들은 다음과 같다.

$b'_{n,i}(x)=B_n(f_n^i(x))$, 단, $x \in QR_n$에 대해서
$B_n(x)=par(x)$

위 식을 이용해서 모든 n, x에 대해서 $b'_{n,i}(x)=b_{n,i}(x)$임을 보이는 것은 용이하다. QRGEN에 의해서 생성된 비트들이 의사난수인지를 결정하는데 필요한 알고리즘은 다음과 같이 정의할 수 있다.

정의 8 : 다항식 크기 회로 $C=\{C_k : k \geq 1\}$는 다음과 같은 조건을 만족하면 QRGEN에 의해서 생성된 길이 Q(k)인 비트열로 부터 정확히 다음 비트 값을

예측하는데 1/P-advantage를 갖는다.(APR(C, N, Q, 1/2+1/P))

조건) $k \in I$인 유한 지수들을 제외한 모든 지수들에 대해서 아래의 특성을 갖는다. $Pr[C_k(b_{n,0}(x), \cdots, b_{n,Q(k)-1}(x))=b_{n,-1}(x)] \geq 1/2+1/P(k)$

단, $n \in N_k$에 대해서 x는 OR_n상에 존재

따라서 모든 다항시 함수 P에 대해서 다음과 같은 식이 성립된다.

$$(\exists C)\ (\exists Q)\ APR(C, N, 1/2+1/P) \Rightarrow (\exists C)\ APAR(C, N, 1/2+1/P) \quad (9)$$

계산 복잡도 관점에서 볼 때 알고리즘 APAR(C, N, 1/2+1/P)는 충분히 큰 두 소수의 곱인 합성수를 소인수 분해하는 문제로 귀착되므로 정리 1이 성립된다.

정리 1. 평방잉여 생성기는 예측 불가능하다. [GM]

4.4 지수 생성기(Index Generator)

충분히 큰 소수 p에 대해서 다음의 식이 주어질 때 $y=g^x \bmod p$에서 x를 구하는 문제, 즉 이산대수 문제는 역시 NP문제이다. 이러한 계산상의 어려움을 근거로 하는 의사난수 생성기인 지수 생성기를 구성하면 다음과 같다.

g는 mod p의 원시원소이고 $x \in QR_p$는 mod p의 임의의 평방잉여이다. 그리고 어떤 정수 $t<(p-1)/2$에 대해서 $index_{p,g}(x)=2t$는 알려져 있다. p, g와 관련해서 x의 principal square root 즉, PQR(p, g, x)는 $g^t \bmod p$이고 nonprincipal square root 즉, NPQR(p, g, r)는 $g^{t+(p-1)/2} \bmod p$이다. 위에서 정의된 p, g에 대해서 predicate $B_{p,g}$는 식 (10)과 같다.

$$B_{p,g}(x)=\begin{cases} 1 & \text{if } x=PQR(p, g, x^2 \bmod p) \\ 0 & \text{if } x=NPQR(p, g, x^2 \bmod p) \end{cases} \quad (10)$$

위의 식을 식(11)으로 표기할 수 있다.

$$B_{p,g}(g^t \bmod p)=\begin{cases} 1 & \text{if } t<(p-1)/2 \\ 0 & \text{if } t \geq (p-1)/2 \end{cases} \quad (11)$$

위 두 식에서의 중요한 특징으로 함수 $B_{p,g}$를 계산할 수 있는 효율적인 알고리즘의 존재는 함수 $index_{p,g}$를 계산할 수 있는 효율적인 알고리즘의 존재를 암시한다. 정리 2는 PQR를 계산할 수 있는 효율적인 알고리즘의 존재는 함수 $index_{p,g}$를 계산할 수 있는 효율적인 알고리즘의 존재를 암시한다.

정리 2. 임의의 소수 p와 원시근 $g \in Z_p^*$ 그리고 $x \in QR_p$에 대해서 A(p, g, x)=PQR(p, g, x)를 다항식 시간, 즉 $|p|$ 내에 계산할 수 있는 알고리즘 A가 존재한다고 가정하자. 이때 다음 식을 다항식 시간, 즉 $|p|$ 내에 계산할 수 있는 알고리즘 A′가 존재한다.

$A'(p, g, x)=index(p, g)(x)$

INDGEN(지수 생성기)은 입력으로 〈p, g, x〉을 받아들이고(단, p는 소수, g는 mod p의 원시근이고, x는 $x \in Z_p^*$), 출력은 $b_{p,g,0}(x)$, $b_{p,g,1}(x)$, …, $b_{p,g,i}(x)$, …인 무한 수열이다. 이러한 무한 수열의 비트들의 생성 방법 및 특성은 다음과 같다.

무한수열 $b_{p,g,0}(x)$, $b_{p,g,1}(x)$, …, $b_{p,g,i}(x)$, …은 다음과 같이 정의된다. 소수 p와 mod p의 원시근이 주어질 때 함수는

$$f_{p,g} : Z_p^* \rightarrow Z_p^* \ : x \rightarrow f_{p,g}(x)=g^x \bmod p \quad (12)$$

이고, 이것의 역함수는 $f_{p,g}^{-1} : Z_p^* \rightarrow Z_p^* \ : x \rightarrow f_{p,g}^{-1}(x)=index_{p,g}(x)$이다.

그리고 함수 $f_{p,g}^i$는 식(13)처럼 정의한다.

$$f_{p,g}^i(x)=\begin{cases} x & \text{if } i=0 \\ f_{p,g}(f_{p,g}^{i-1}(x)) & \text{if } i>0 \\ f_{p,g}^{-1}(f_{p,g}^{i+1}(x)) & \text{if } i<0 \end{cases} \quad (13)$$

이 때 소수 p와 mod p의 원시근 g 그리고 $x \in QR_n$에 대해서 비트들은 다음과 같다.

$$b_{p,g,i}(x)=B_{p,g}(f_{p,g}^i(x)) \quad (14)$$

다음은 INDGEN에 의해서 생성된 비트들이 의사난수인지를 결정하는데 필요한 알고리즘에 대한

정의는 다음과 같다.[BM]

정의 9 : 다항식 크기 회로 $C=\{C_k : k \geq 1\}$는 다음과 같은 조건을 만족하면 family N에 대한 함수 $B_{p,g}$를 계산하는데 1/P-advantage를 갖는다.

(AB(C, N, 1/2+1/P)

조건) $k \in I$인 유한 지수들을 제외한 모든 지수들에 대해서 다음의 특성을 갖는다.

$Pr[x \in QR_p : C_k(p, g, x) = B_{p,g}(x)] \geq 1/2+1/P(k)$

단, $p \in N_k$ 그리고 mod p상의 원시근 g

정의 10 : 다항식 크기 회로 $C=\{C_k : k \geq 1\}$는 다음과 같은 조건을 만족하면 INDGEN에 의해서 생성된 길이 Q(k)인 비트열로 부터 정확히 다음 비트 값을 예측하는데 1/P-advantage를 갖는다.(APR(C,N,Q, 1/2+1/P))

조건) $k \in I$인 유한 지수들을 제외한 모든 지수들에 대해서 다음의 특성을 갖는다.

$Pr[C_k(b_{p,g,1}(x), \cdots, b_{p,g,Q(k)-1}(x)) = b_{p,g,Q(k)-0}(x)] \geq 1/2+1/P(k)$

단, $p \in N_k$이고 mod p의 모든 원시근 g

따라서 모든 다항식 함수 P에 대해서 식(15)가 성립된다.

$$(\exists C)\ (\exists Q)\ APR(C, N, 1/2+1/P) \Rightarrow (\exists C)\ AB(C, N, 1/2+1/P) \quad (15)$$

계산 복잡도 관점에서 볼 때 알고리즘 $\overline{AB(C, N, 1/2+1/P)}$는 이산대수 문제로 귀착되므로 정리 3이 성립된다.

정리 3. 지수 생성기는 예측 불가능하다.[BM]

5. 암호에서의 난수

3, 4장에서 열거한 의사난수 생성기들에 의해서 생성된 의사난수가 랜덤한가 하는 랜덤성 여부를 판정하는 방법으로서는 빈도 검정, 독립성 검정, Entorpy 관점에서의 검정법 그리고 포커 검정, 충돌 검정, 자동상관 검정, 연 검정, 쿠폰수집가 검정, 갭 검정, 순열 검정 그리고 최대값 검정 등이 있다.[K] 이와같은 여러가지의 검정을 해야하는 이유로는 임의의 한검정을 통과하였다라는 것은 랜덤하지 않다는 증거를 발견하지 못했다는 것을 의미하지 랜덤하다고는 단정하지 못하기 때문이다. 예로써 middle square method에 의하여 만들어진 의사난수는 frequencyl-test, gap test, poker test는 통과하나 frequency2-test는 통과하지 못한다. 따라서 본 장에서 사용할 의사난수는 위의 검정법들을 모두 통과한 의사난수라고 가정한다.

난수(의사난수)와 의사난수 생성기는 크립토시스템의 여러 분야에서 많이 적용되고 있다. 그 예로서 키 용법에서는 관용키를 의사난수 생성기의 seed로 사용함으로써 의사난수가 시스템의 안전성을 제공하였으며, 키 은닉 용법에서는 난수가 일종의 비밀키로 사용되어 이것을 입력으로 하는 의사난수 생성기에 의해서 생성되는 난수가 위의 비밀키를 은닉하는 방식이다.[GM][GGM][LR] 데이타 변환 용법의 한 예인 Bit-commitment 방식은 의사난수 생성기에 의해서 생성된 의사난수 수열을 사용하여 자신이 생성한 데이타를 변환하여 상대방에게 보내는데 상대방은 자신의 데이타를 예측할 수 없으며, 또한 자신은 자신의 데이타를 부정할 수 없는 방식이다. 즉, m번째 비트의 값이 주어진 상태에서 다음 비트를 예측하려고 시도하는 임의의 다항식시간 알고리즘은 1/2+1/p(n) 보다 적은 성공확률을 갖는다라는 특성을 이용한다.[No] 또한 의사난수 생성기와 난수를 이용한 영지식 상호증명 방식인 Fiat-Shamir 개인식별 방식(이하 FS 방식)에서는 의사난수 생성기에 의해서 출력되는 수열들을 이용하여 비밀키를 생성하고, 영지식이 성립되기 위해서 필요한 난수는 동전던지기(coin-flipping)방식을 이용하여 생성하였다.[FFS][B] 그리고 mu-

lti-party 프로토콜에서는 각 참가자의 비밀입력에 대한 정보를 숨기기 위해서 의사난수 생성기를 사용한다.[BG]

본 장에서는 의사난수 생성기의 적용사례별로 분류하여 기술하고자 한다.

5.1 키 용법

본 절에서는 관용키를 의사난수 생성기의 seed로 사용함으로써 의사난수가 시스템의 안전성을 제공하는 방식에 대해서 고찰하고자 한다.

우선 A와 B가 안전한 통신을 원한다고 가정하자. 의사난수 생성기가 존재한다는 가정하에 이 두 참여자를 위한 관용키 암호 시스템은 다음과 같다.

단계 1) 관용키는 의사난수 생성기의 입력으로써 랜덤하게 선택된 seed이다.

단계 2) 암호화는 메세지를 의사난수 생성기에 생성된 스트링들과 XORing 함으로써 행해진다.(단, 같은 스트링들은 사용될 수 없다. 즉, 의사난수 생성기의 출력은 "one-time pad"로써 사용한다.)

단계 3) 복호화는 암호문을 의사난수 pad와 같은 스트링을 가지고서 XORing함으로써 복호된다.(실제응용에 있어서 암호문들은 순서대로 나타나지 않기 때문에 의사난수 pad의 순서는 표시되어야 한다.)

위의 시스템은 one-time pad가 명백히 랜덤하다면 이 시스템은 안전하다. 그러므로 제안된 방식의 비안전도(insecurity)는 의사난수 스트링들과 랜덤 스트링들을 구별하는 검사로 확인할 수 있다.(모순)

Note : 만약 A와 B가 메세지들의 길이를 프로토콜 시작전에 제한하는 것을 원치 않는다면 그들은 무한한 비트를 출력하는 의사난수 생성기를 사용해야 한다. 물론 이러한 의사난수 생성기는 더이상 다항식 시간 bound가 아니다. 따라서 효율성과 안전성에 대한 관계를 잘 설정해야 한다. 흥미있는 사실은 다음 비트를 생성하기 위해서 소요되는 시간은 seed의 길이와 다항식 함수값 만큼 비례한다.

5.2 키 은닉 용법

본 절에서는 난수가 일종의 비밀키로 사용되며, 이러한 비밀키를 seed로 하는 의사난수 생성기에 의해 생성되는 난수를 이용하여 위의 비밀키 은닉 방식에 대해서 고찰하고자 한다. 소인수 분해문제는 NP문제라는 가정하에서 난수와 의사난수 생성기를 이용한 공개키 암호시스템은 다음과 같다.

단계 1) 이 시스템은 다항식 시간 확률적 알고리즘인 (G, E, D)로 구성되고 $G(1^n)$은 2개의 큰 소수 p, q를 생성한다.(단, $p \equiv q \equiv 3 \bmod 4$, $N=p \cdot q$)

단계 2) N은 공개키로서 사용하고 그것의 소인수 분해된 소수들은 비밀키이다. (일반적으로 G는 함수와 함정성(trapdoor)에 대한 지표를 생성한다.)

단계 3) 메세지 $\beta=\beta_1 \cdots \beta_m$을 암호화하기 위해서, 암호화 알고리즘(E)는 mod N상에서 랜덤 평방 잉여인 X_0를 선택해서 다음 식을 계산한다.

$$X_i \leftarrow X_{i-1}^2 \bmod N$$
$$\sigma_i \leftarrow LSB(X_{i-1})$$

단, LSB(x) : mod N상에서의 제곱과 관련한 hard-core 비트이다.

단계 4) $E(\beta)$의 출력은 X_n과 $(b_1 \cdots b_m)$이다.

단, $b_1 \cdots b_m$은 원래의 메세지와 X_0를 seed로써 사용한 의사난수 생성기의 출력과 XORing한 결과값이다. 즉, $b_i=\sigma_i \oplus \beta_i$ 암호화된 값의 길이는 m+n이다.

단계 5) N을 구성한 2개의 소수를 아는 상태에서 복호화 알고리즘(decipering algorithm) $D(\beta)$은 반복해서 X_m의 평방근을 발견한 다음 seed를 구해서

암호화된 메세지를 복호화한다.

(Blum integers의 중요한 특성이 여기서도 사용된다. 즉, 만약 p와 q가 blum integer이면, mod p·q의 평방잉여근들 중의 오직 하나만이 평방잉여이다. 따라서 X_m으로 부터 X_0를 구할 수 있다.)

만약 G가 $\sigma_1 \cdots \sigma_m$을 출력할 때 G는 의사난수 생성기다라는 사실은 알려져 있지만, 여기서는 G가 X_m을 출력한다. 따라서 이 방식이 타당하기 위해서는 다음의 정리가 성립되어야 한다.

정리 6. 만약 G가 역시 X_m을 출력하더라도 생성기 G는 의사난수 생성기이다.

5.3 데이타 변환 용법

의사난수 생성기에 의해서 생성된 의사난수 수열은 다음 비트의 값에 대한 예측할 수 없게 하는 특성을 갖는다. 즉, 의사난수 수열의 m번째 비트의 값이 주어진 상태에서 다음 비트를 예측하려고 시도하는 임의의 다항식시간 알고리즘은 $1/2+1/p(n)$ 보다 적은 성공확률을 갖는다. 본 절에서는 bit-commitment를 얻기 위해서 이러한 특성을 적용한다.

Bit-commitment 방식을 구성하기 위한 첫번째 시도로서 다음의 프로토콜을 생각해 보자.[BCC]

(1) Commit 단계 :

Alice는 seed $s \in \{0, 1\}^n$을 선택하고, $G_m(s)$와 $B_{m+1}(s) \oplus b$를 보낸다.

(2) Reveal 단계 :

Alice는 s를 보내고, Bob은 $G_m(s)$가 앞단계에서 자기에게 보낸 것인지 확인하고, $b=B_{m+1} \oplus (B_{m+1}(s) \oplus b)$을 계산한다.

단, b : Alice가 commit할려고 하는 비트

$G_m(s)$: seed s을 이용 의사난수 생성기에서 생성된 수열에서의 첫 m개의 비트들

$B_{m+1}(s)$: seed s에 대한 의사난수 수열의 m+1번째 비트

위의 프로토콜은 Alice가 commit 할려고 하는 비트 b를 $1/2+1/poly(n)$ 보다 큰 확률로 Bob이 예측할 수 없다는 특성을 갖는다. 왜냐하면 Bob은 의사난수 수열을 예측할 수 있는 능력을 갖지 못하기 때문이다. 반면에 Alice는 Bob을 속일 수 있을런지 모른다. 만약 Alice가 $G_m(s_1)=G_m(s_2)$이고, $B_{m+1}(s_1) \neq B_{m+1}(s_2)$인 두개의 seed들, 즉 s_1과 s_2를 발견했다면 Alice는 자신이 원하는 임의의 비트를 밝힐 수 있다. 의사난수 생성기의 정의에 의해서 그러한 쌍의 존재를 금지하는 것은 없다. 그러므로 임의의 의사난수 생성기 G가 주어지면 그러한 쌍을 가지는 다른 의사난수 생성기 G′가 구성될 수 있다.

같은 수열을 생성하는 두개의 seeds가 존재할 수 있기 때문에 Alice를 하나의 seed만 사용하도록 하는 방법은 없다. 그러므로 다음 프로토콜의 목적은 Alice가 같은 의사난수 수열만을 사용하도록 하는 것이고, 혹은 Alice가 높은 확률로 속이는 행위가 발견되도록 하는 것이다.

Bit-commitment 프로토콜

(1) Commit 단계

a) Bob은 랜덤벡터 $R=(r_1, r_2, \cdots, r_{3n})$을 선택하고 그리고 그것을 Alice에게 보낸다.

단, r_i : $r_i \in \{0, 1\}$ for $1 \le i \le 3n$

b) Alice는 하나의 seed $s \in \{0, 1\}^n$을 선택하고 벡터 $D=(d_1, d_2, \cdots, d_{3n})$을 Bob에게 보낸다. 단,

$$d_i = \begin{cases} b_i(s) & \text{if } r_i=0 \\ B_i(s) \oplus b & \text{if } r_i=1 \end{cases} \qquad (16)$$

(2) Reveal 단계

Alice는 s를 보내고 Bob은 모든 $1 \le i \le 3n$에 대해서

만약 $r_i=0$이면 $d_i=B_i(s)$인지를 검사하고,

만약 $r_i=1$이면 $d_i=B_i(s) \oplus b$인지를 검사한다.

이러한 프로토콜은 Bob이 비트 b에 대한 정보를 얻을 수 없다는 특성을 유지한다. 그렇지 않다면 Bob은 의사난수 생성기의 출력과 완전한 랜덤 스트링을 구별할 수 있는 능력을 갖고 있다는 것을 의미한다. 즉, 만약 Alice가 의사난수 수열 대신에 완전한 랜덤 수열을 선택했더라도, 그때 Bob은 b에 대한 어떠한 정보도 얻지못할 것이다. 왜냐하면 모든 벡터 D는 b가 무엇이든지 간에 같아 보일 것이기 때문이다.(이것은 확률 $q>1/2$로 Bob이 b를 예측하는 것을 허용하는 어떤 임의의 입력을 Bob이 가진 일반적인 경우에서도 사실이다.) 만약 Alice가 의사난수 수열을 사용했을 때 b에 대한 어떤 정보를 배울 수 있는 확률적 다항식시간 Bob(이때 bob을 Bob'로 칭함)이 존재한다면, 그때 Bob'는 G의 출력과 완전한 랜덤수열의 차이를 구별할 수 있는distingusher를 구성하는데 사용될 수 있다. 수열 x가 주어지고 Alice와 Bob'가 프로토콜의 commit단계를 실행했다고 가정하자.(단, Alice는 랜덤비트 b를 commit하고 의사난수 수열을 생성하는데 대신에 x를 사용) 이때 Bob'가 b를 예측하려고 한다고 하자. 만약 그가 옳바르게 추측했다면 x는 의사난수라고 결정할 것이고 그렇지 않으면, x는 완전한 랜덤이라고 결정할 것이다. 완전한 랜덤수열과 의사난수 수열사이에서 그 수열이 의사난수라고 결정할 수 있는 확률과 완전한 랜덤이라고 결정할 수 있는 확률과의 차이가 의미하는 것은 x가 의사난수 수열인 경우에서 Bob'가 b를 예측하는데 가지 수 있는 이점을 말한다.

Alice가 어떻게 해서 Bob을 속일 수 있는가? 그녀가 속일 수 있는 기회는 $r_i=0$인 모든 i에서 $G_{3n}(s_1)$과 $G_{3n}(s_2)$가 같고, $r_i=1$인 모든 i에 다른 두개의 seeds s_1과 s_2가 존재하여야 한다. 따라서 그러한 쌍은 R을 쓸모없는 것으로 만들것이다.

정리 7. R을 쓸모없는 것으로 만드는 s_1과 s_2의 한쌍이 존재하는 확률은 최대 2^{-n}이다.

〈증명〉 만약 s_1과 s_2의 한쌍이 R을 쓸모없는 것으로 만든다면, 그때 $r_i=B_i=(s_1)\oplus B_i(s_2)$을 알 수 있다. 그러므로 s_1과 s_2는 정확하게 하나의 R을 쓸모없는 것으로 만든다. 프로토콜에서 2^{2n}개의 seed쌍들과 2^{3n} 벡터 R이 있다. 따라서 Bob이 선택할 벡터 R을 쓸모없는 것으로 만들 수 있는 seed쌍이 존재하는 확률은 최대 $2^{2n}/2^{3n}=2^{-n}$이다.

정리 8. G가 의사난수 생성기라면, 위에서 제안한 bit-commitemt 프로토콜은 다음의 특성을 갖는다.

모든 다항식 p와 충분히 큰 securiy parameter n에 대해서

a) commit단계를 수행하는 확률적 다항식시간 Bob은 $1/2+1/p(n)$ 보다 더 큰 확률로 b를 예측할 수 없다.

b) Alice는 2^{-n} 보다 적은 확률로 오직 하나의 가능한 비트를 밝힐 수 있다.[No]

5.4 기타 용법

앞 절에서 설명한 용법 이외에도 효율적인 확률적 알고리즘의 구성, error-correcting code 그리고 영지식 상호증명 방식과 같은 기타 용법이 있다.

본 절에서는 의사난수 생성기와 난수를 이용한 영지식 상호증명 방식인 FS 방식을 설명하고자 한다. FS 방식에서는 의사난수 생성기에 의해서 출력되는 수열들을 이용하여 비밀키를 생성하고, 영지식이 성립되기 위해서 필요한 난수는 동전던지기(coin-flipping) 방식을 이용하여 생성하였다.

우선 개인 식별 정보 ID_i의 평방잉여 s_i를 계산하여 가입자의 비밀키로 사용하였다. 이 방식의 안전성은 충분히 큰 두 소수 p, q의 곱인 n의 소인수분해를 모를 때, 제곱근을 구하는 문제는 어려운 문제(NP 문제)라는 것에 근거한다.

사전 준비 과정에서 신뢰할 수 있는 센터는 소수 p, q를 비밀리에 선택하고, 그 곱인 n을 공개한다. 카드발급 과정에서 센터는 합법적인 사용자에게 카드를 발급할 때 그 사용자에 관한 정보(이름, id번호, 주소, 주민등록번호 등)와 카드에 관한 정보(유효기간 등)를 담고 있는 ID_i를 준비하고, mod n상에서 ID_i의 평방근을 계산하여 그 역수 s_i를 각 가입자의

비밀키로 한다.

즉, 이 방식에서는 모든 ID_i가 mod n상에서 평방근을 갖지는 않으므로, 이 문제를 해결 하기 위해 임의의 스트링이 입력되면[0, n)이 출력되는 의사난수 생성기 h를 선택하여 공개하고 아래와 같이 비밀키를 생성한다.

① $v_j=h(ID_i, k_j)$ (j=1, …, m)을 구한다.

② 이 중에서 k개의 평방잉여 선택, 각 v_j의 가장 작은 제곱근 s_j를 k개 계산.

③ 1와 k개의 s_j, 그리고 각각의 j값을 카드에 담아 사용자에게 발급.

가입자 A와 가입자 B가 개인 식별을 행하는 프로토콜은 아래와 같다.

개인 식별 프로토콜

단계 1) 가입자 A는 ID_A를 가입자 B에게 전송한다.

단계 2) 가입자 B는 $v_j=h(ID_A, k_j)$ (j=1, …, k)를 계산한다.

단계 3-1) 가입자 A는 $r \in_R Z_n^*$ 를 선택한다.

3-2) 가입자 A는 $x=r^2 \pmod n$을 계산한다.

3-3) 가입자 A는 x를 가입자 B에게 전송한다.

단계 4-1) 가입자 B는 난수 $(d_1, \cdots, d_k) \in_R \{0, 1\}$를 선택한다.

4-2) 가입자 B는 가입자 A에게 $(d_1, \cdots, d_k)$를 전송한다.

단계 5-1) 가입자 A는 $Y=r \prod_{d_j==1} s_j \pmod n$을 계산한다.

5-2) 가입자 A는 Y를 가입자 B에게 전송한다.

단계 6) 가입자 B는 $x=y^2 \prod_{d_j==1} v_j \pmod n$인지 검증한다.

단계 7) 순서 3-1에서 순서 6을 t회 반복한다.

위의 FS 방식은 영지식이며 안전성은 동전던지기 방식에 의해서 생성된 난수의 갯수 k와 매개 변수(security parameter) t에 의존한다(2^{-kt}).

FS 방식에서 B의 역할은 수동적이나 매우 중요하다. 즉, B가 전송한 난수 d_j는 어떠한 정보도 포함하고 있지 않고, 그 난수의 비예측성(unpredictability)은 A가 속이려고 하는 시도를 방지한다. 그리고 사전 준비 과정에서 신뢰할 수 있는 센터가 선택한 의사난수 생성기 h 역시 안전성 및 collision-free를 제공한다.

다음은 FS 방식의 실제 예이다.

사전 단계) I={10111110100011100}, (단, I는 이름, id번호, 주소, 주민등록번호등을 포함)

n=35, k=5.

① $v_j=h(I, j)$ (j=1, …, 15)을 구한다.

② 이 중에서 5개의 평방잉여를 선택, 각 v_j의 가장 작은 제곱근 s_j를 5개 계산.

③ I와 5개의 s_j, 그리고 각각의 j값을 카드에 담아 사용자에게 발급. 즉, 1 4 6 3 7 8 8 4 11 1 12 혹은 1 s_6 6 s_7 7 s_8 8 s_{11} 11 s_{12} 12

단계 1) 가입자 A는 I를 가입자 B에게 전송한다.

단계 2) 가입자 B는 $v_j=h(I, j)$ (j=6, 7, 8, 11, 12)를 계산한다.

단계 3-1) 가입자 A는 $r_1 \in Z[0, 35)$를 선택한다.

3-2) 가입자 A는 $x_1=6^2 \pmod n=1$을 계산한다.

3-3) 가입자 A는 x_1를 가입자 B에게 전송한다.

단계 4-1) 가입자 B는 난수 (10111)를 선택한다.

4-2) 가입자 B는 가입자 A에게 (10111)를 전송한다.

단계 5-1) 가입자 A는 $Y=\prod_{d_j=1} s_j \pmod{35}$
$=6 \cdot s_1 \cdot s_3 \cdot s_4 \cdot s_5 \pmod{35}$
$=6 \cdot 4 \cdot 8 \cdot 4 \cdot 1 \pmod{35}$
$=33 \pmod{35}$을 계산한다.

5-2) 가입자 A는 Y_1를 가입자 B에게 전송한다.

단계 6) 가입자 B는 $x_1=Y\prod_{d_j=1} v_j \pmod{35}$
$=33 \cdot 11 \cdot 29 \cdot 11 \cdot 1 \pmod{35}$
$=1 \pmod{35}$인지 검증한다.

단계 7) 단계 3에서 단계 6을 t회 반복한다.

6. 결 론

정보화 사회의 진전으로 컴퓨터 통신의 활용이 모든 분야에서 필수적으로 대두될 것이며, 이에 따라 전송중인 데이타에 대한 보안대책이 필수 불가결할 것이다. 이러한 데이타에 대한 보안 대책으로 가장 효과적인 방법으로써 암호화 기법이 이용되고 있다. 암호화 기법에는 여러 종류가 있겠으나 거의 모든 암호 함수는 의사난수를 사용하고 있다. 이러한 의사난수는 암호학에 있어서 매우 중요한 역할을 하고 있다.

본고에서는 의사난수에 대한 정의를 구별 불가능 개념과 예측불가능 분포 개념을 사용하여 고찰하였으며, 이에 따른 의사난수 생성기의 구성에 관해 고찰하였다. 그리고 난수 및 의사난수 생성기를 사용 용법별(기 용법, 키 은닉 용법, 데이타 변환 용법 그리고 기타용법 등)로 분류하여 난수와 암호와의 관계를 고찰해 보았다.

따라서 본고에서 고찰한 결과를 이용하여 난수의 체계적인 연구와 난수를 이용한 암호 시스템의 연구에 일조가 되길 기대한다.

참 고 문 헌

[B] M. Blum, "Coin flipping by Telephone," Proc. 24th IEEE Compcon, pp. 133-127, 1982.

[BBS] L. Blum, M. Blum, and M. Shub, "A Simple Secure Pseudorandom Generator," IEEE Crypto 82, 1982.

[BCC] G. Brassard, D. Chaum, and C. Crepeau. "Minimum Disclosure Proofs of Knowledge," JCSS, Vol. 37, No. 2, pp. 156-189, 1988.

[BG] D. Beaver, and S. Goldwasser, "Multiparty Computation with Faulty Marjority," Proc. 30th FOCS, IEEE, pp. 468-473, 1989.

[BM] M. Blum, and S. Micali, "How to Generate Cryptographically Strong Sequences of Pseudorandom Bits," 23rd IEEE FOCS pp. 112-117, 1982.

[Choi] 최봉대, "Randomness 특성분석에 관한 연구" 데이타 보호의 기반 기술 연구 최종 평가회 자료집, 1991.

[FFS] U. Feige, A. Fiat, and A. Shamir. "Zero knowledge Proofs of identity," The 19th ACM STOC pp. 210-217, 1988.

[Go] O. goldreich, "Foundation of Cryptography," Class Notes pp. 88-95, 1989.

[GM] S. Goldwasser, and S. Micali, "Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information," 14th ACM STOC, pp. 165-177, 1982.

[GMR] S. Goldwasser, S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proof Systems," The 16th ACM STOC, pp. 291-304, 1985.

[H] J. Hastad, "Pseudorandom Generators under uniform assumptions," proc. 22th ACM STOC, pp.395-404, 1990.

[K] D.E. Knuth, "The Art of Computer Programming : Seminumerical Algorithm," Addison-Wesley, Vol.11, 1981.

[L] L.Levin, "One Way Functions and Pseudorandom Generators," The 17th ACM, pp.363-365. 1985.

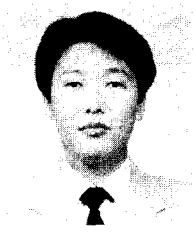
[M] M.Mignotte, "Tests de Primalite," Theor. Computer Science(12), 1980.

[MH] R.Merkle, and M.Hellman, "Hiding Information and Signatures in Trapdoor Knapsacks," IEEE Transactions on Information Theory, IT.24-5, 1978.

[No] M.Noar, "Bit Commitment Using Pseudorandomness," Journal of Cryptolgy, pp.151-158, 1991.

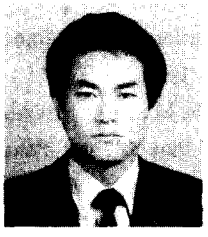
[Y] A.C.Yao, "Theory and Application of Trapdoor Functions" Proc. 23rd FOCS, IEEE, pp.80-91, 1982.

□ 著者紹介



양 형 규(정회원)

1983년 성균관대학교 전자공학과 졸업(공학사)
1985년 성균관대학교 대학원 전자공학과 졸업(공학석사)
1991년~현재 성균관대학교 정보공학과 박사과정 재학중
1985년~1991년 삼성전자 컴퓨터부문 선임 연구원



안 영 화(종신회원)

1975년 성균관대학교 전자공학과 졸업(공학사)
1977년 성균관대학교 대학원 전자공학과 졸업(공학석사)
1990년 성균관대학교 대학원 전자공학과 졸업(공학박사)
1983년 5월~1990년 2월 해군사관학교 전자공학과 조교수
1990년 3월~현재 강남대학교 전자계산학과 조교수