[UCI]I804:24011-200000256819

2012年 2月
博士學位論文

양자암호 시스템에서 의사난수 사용에 대한 취약성 개선방법

朝鮮大學校 大學院

電子計算學科

崔 眞 錫

2012年2月 博士學位論文 양자암호 시스템에서 의사난수 사용에 대한 취약성 개선방법 崔 眞 錫

양자암호 시스템에서 의사난수 사용에 대한 취약성 개선방법

An Improvement Method of Vulnerability using Pseudo Random Number of Quantum Cryptography

2012年 2月 24日

朝鮮大學校 大學院

電子計算學科

崔 眞 錫

양자암호 시스템에서 의사난수 사용에 대한 취약성 개선방법

指導敎授　李　聖　周

이 論文을 理學博士學位申請 論文으로 提出함

2011年 10月

朝鮮大學校 大學院

電子計算學科

崔　眞　錫

崔眞錫의 博士學位 論文을 認准함

委員長 光州女子大學校 敎 授 정 성 옥 (인)

委 員 朝鮮大學校 敎 授 조 범 준 (인)

委 員 朝鮮大學校 敎 授 배 용 근 (인)

委 員 南部大學校 副敎授 정 병 수 (인)

委 員 朝鮮大學校 敎 授 이 성 주 (인)

2011年 12月

朝鮮大學校 大學院

목 차

표 목 차

그림목차

ABSTRACT

An Improvement Method of Vulnerability using Pseudo Random Number of Quantum Cryptography

Choi Jin-Suk
Advisor : Prof. Lee Sung-Joo, Ph.D.
Department of Computer Science,
Graduate School of Chosun University

In this thesis, quantum cryptography systems used in the design process inevitably open bit stream of pseudo-random number that exists multiple open channels between them and the need to share information on the part of the situation exposes a pair of bit stream. and, randomness is a basic security evaluation item for the most cryptographic algorithms.

NIST(National Institute of Standards and Technology) has proposed a statistical test suit for random number generators for cryptographic applications in the process of AES(Advanced Encryption Standard) project. However the test suit of NIST(National Institute of Standards and

Technology) is customized to block ciphers which have the same input and output lengths.

And, we propose the base test of pseudo-random number I tested out this process and the merge bit binary column look out for randomness, quantum photons generators by the true random number using quantum cryptography algorithm propose.

Also, the resulting vector which is also used and generated real random number by applying the principles of quantum theory, quantum bits, showed the result of a random number of causes.

In particular, BB84 protocol an improvement over the double and single-photon generator of single-photon generator of the difference occurs in the gaps in the data flow but the cost and hardware for implementing true random number inconvenience caused random number sequence to improve the process quantum bit is applied.

The proposed double and single-photon generator of pseudo-code algorithm is applied using multiple real random number be able to take as a reference is considered.

Ⅰ. 서 론

A. 개요

인터넷과 컴퓨터의 발달로 정보 통신망이 급속히 보급됨에 따라 공중통신망을 통한 정보 전송이 급격히 증가하면서 정보 보호에 따른 안전한 암호 알고리즘이 필요하게 되었다.

암호학의 발달 과정은 고전에는 주로 단순한 문자를 대입하고 이를 치환하여 정보를 은닉하였다. 이는 문장에 나타나는 문자들의 통계적 특성을 그대로 나타내기에 암호문의 통계적 특성을 분석하여 암호문을 해독할 수 있었다. 이후 복잡한 기계를 이용하여 암호문을 작성하였는데 이를 해독하기 위해서는 많은 양의 계산을 필요로 하였기에 안전한 시스템이라 볼 수 있었다. 그러나 현대와 같이 정보전산능력이 뛰어난 시기에는 그도 또한 쉽게 해독할 수 있다. 이후 Shannon에 의해 복잡도가 높은 암호 알고리즘의 실현이 가능하게 되었고 현대 암호학의 기본 토대를 형성하게 되었다[20,41].

암호는 크게 대칭키 암호 시스템과 공개키 암호 시스템으로 나눌 수가 있다. 1970년대 초 Shannon에 의해 주장된 혼돈(Confusion)과 확산(Diffusion)을 여러번 반복하면 강력한 암호 알고리즘을 구현할 수 있다는 이론에 의해 미국의 표준암호 알고리즘인 미국 상무성 표준국(NBS : National Bureau of Standard 후에 NIST : National Institute of Standards and Technology)은 Brooks ACT 89-306에 따라 암호 표준화 연구를 시작하였다. 미국 상무성은 표준 암호 알고리즘은 높은 수준의 안전성을 보장할 수 있어야 하며, 사양의 정의가 완전하여 간단히 이해할 수 있어야 하며, 알고리즘의 비밀성에 의존 되어서는 안되며, 사용자나 제작자가 모두 사용 가능해야 하며, 표준 암호 알고리즘의 응용이 다양해야 하며, 전자 장치로써 제품화가 간단하고 또한, 사용이 간단해야 하며, 알고리즘 타당성 검증에 협력해야 하며, 표준 암호 알고리즘은 수출할 수 있어야 한다는 전제로 표준 암호 알고리즘을 공모하여 DES(Data Encryption Standard)가 IBM에 의해 제안되어 많은 기간 사용되었으며 이후 AES로 발전하였다[15,17,18,24,25,26,29,40].

원타임패드(One-time pad)와 유사한 안전성을 보장하며 일반적인 통신망에도 적용할 수 있는 암호 알고리즘으로 스트림 암호 시스템이 있는데 이는 난수를 생성하

여 평문과 일대 일로 대칭하여 암복호화하는 방식이다. 이것 또한 엄밀한 의미에서 대칭키 암호 알고리즘으로 볼 수 있다. 대칭키 암호 알고리즘은 일단 키를 송신자와 수신자가 똑같이 나누어 가져야 한다는 불편이 있다. 이를 해소하기 위하여 암호화와 복호화 과정에서 서로 다른 키를 사용하고 암호화 키를 공개하여 키의 전송 및 비밀 보관 등이 필요없게 만든 것이 공개키 암호 시스템이다. 이는 1976년 Diffe와 Hellman의 연구[43]에 발표가 되었다. 이도 또한 발전을 거쳐서 현재 RSA, Elgamal, 타원곡선암호, 땋임군 암호 등이 나와 있으나 계산량이 너무 많기 때문에 일반 평문에 대한 암호화는 힘들고 주로 키 분배 알고리즘과 짧은 길이의 데이타에 대해 사용되고 있는 실정이다[3,8,14,19,23,27].

의사난수 발생기는 거의 모든 암호학적 알고리즘에서 골격과 같이 사용되는 가장 중요한 암호학적 함수 중의 하나이다. 기존 의사난수 생성 알고리즘은 대수학적 이론에 기반을 둔 것과 하드웨어적 알고리즘에 기반을 둔 것으로 볼 수 있다. 의사난수 생성기는 스트림 암호의 원천을 이루고 또한 암호 프로토콜의 초기 벡터 또는 비밀 키, 전자서명 및 전자결제 시스템의 비밀 파라미터, 각종 키 관리/인증 메커니즘에서의 세션 키 생성 등에 사용된다. 난수는 실난수(True Random)와 의사난수(Pseudo Random)[36,37,39,42]로 나눌 수 있는데 그 차이점은 실난수는 비결정적임이며 예측 불가능한 어떤 물리적인 소스로부터 획득되며, 예로서 저항소자, 반도체 다이오드, 방사능물질로부터의 전기적인 잡음을 난수발생의 근거로 삼는다. 의사난수는 컴퓨터는 논리적이고 결정적 이므로 실난수를 산출 하지 못하며 S/W에 기반한 RNG는 최상의 경우에 의사난수를 생성 가능 하며 PRNG 는 길이가 짧은 랜덤 비트 열(Seed)을 길이가 긴 랜덤에 근접한 비트열로 출력하는 알고리즘이다.

난수성을 만족하는 이상적인 난수의 특징은 긴 주기를 가져야 하며 패턴과 역상관 관계를 알 수 없어야 하고 1과 0의 평균 생성횟수가 동일하여야 한다. 그리고 “01010101”과 같은 비트 스트링의 형태가 길지 않아야 하고 간단한 알고리즘으로 설계와 구현이 쉬워야 한다. 마지막으로 난수의 특징은 이미 나온 출력으로부터 그 전의 값을 유추할 수 없어야 한다. 이러한 난수성을 만족하고 구현상의 용이함을 위하여 암호학적 알고리즘을 이용한 의사난수 생성기를 사용한다.

암호학적으로 안전한 의사난수 생성기는 Blum, Micali에 의해서 처음으로 소개가 되었다. 그 이후로 다양한 의사난수 생성기가 제안되었다. Goldreich, Goldwasser, Micali는 의사난수 생성기를 사용해서 의사 랜덤 함수 생성기(Pseudo Random Function Generator)를 만드는 것이 가능함을 보였다. 그 뒤 Luby, Rackoff는 의사 랜

덤 함수 생성기(Pseudo Random Function Generator)와 DES 구조를 사용해서 의사 랜덤 순열 생성기(Pseudo Random Permutation Generator)를 만들 수 있음을 보였다.

다음은 암호학적으로 강력한 의사난수 생성기로 알려진 ANSI X9.17[5,12]은 ANSI가 추천하는 난수 생성 방식으로 암호학 응용분야에 널리 사용되고 있다. ANSI X9.17 난수 생성기는 세 개의 Triple DES를 사용하여 구성된다. ANSI X9.17 난수 생성기의 입력으로 날짜와 시간을 나타내는 64비트의 DT_i 와 64비트의 시드(Seed)초기 값 V_i가 입력된다. 세 개의 Triple DES의 키 입력으로 두 개의 54비트 K_1, K_2가 입력되면 ANSI X9.17 생성기는 64비트의 난수 V_{i+1}를 출력한다. 따라서 ANSI X9.17 난수 생성기의 출력 난수는 두 개의 입력 DT_i와 V_i에 의해 식 (1.1)이 결정된다.

$$\text{출력난수 : } R_i = EDE_{K_1,K_2}(V_i \oplus EDI_{K_1,K_2}(DT_i)) \tag{1.1}$$

$$\text{다음 시드(seed)값 : } V_{i+1} = EDE_{K_1,K_2}(R_i \oplus EDI_{K_1,K_2}(DT_i)) \tag{1.2}$$

본 논문에서는 양자암호 시스템 내에서 사용되고 있는 난수열들에 대하여 의사난수를 이용하는 구간을 분석하고 이에 대한 이진 병합적 접근으로 부분정보를 유추할 수 있는 방법에 대하여 언급 한다. 원타임패드(One-time pad)의 안정성에 의하여 절대적 도청불가 상태의 암호 구현을 실현한 양자암호 시스템 내에서 사용자의 실난수 사용에 대한 불편함으로 인하여 의사난수를 사용한다. 이는 단일 비트열에서는 입증된 난수성이 다중 비트열의 병합에 의하여 부분 정보나 난수성의 오류를 보이는 것을 알 수 있다.

B. 연구배경 및 연구목적

정보를 비밀리에 보내는 것과 관계되는 암호화 작업은 신용카드의 비밀 번호에서부터 컴퓨터의 패스워드까지 이미 사용되어 오고 있다. 현재까지 주로 사용되는 고전적 방식의 암호화는 엄밀히 말하여 완벽한 안전을 보장하는 것이 아니라 현실적으로 풀기에는 많은 시간과 자원이 필요하다는 조건부 안전에 의존하고 있다. 그러나 빠른 컴퓨터의 개발과 수학적 이론의 발달은 안전의 커다란 위협요소이다. 세상

에 존재하는 모든 도청 방법에서 자유로운 완벽한 비밀을 보장해주는 방법은 도청하려는 사람이 인간이 할 수 있는 모든 기술을 동원하리라는 전제하에 자연 법칙 그 자체가 안전을 보장해줌을 의미한다. 안전하게 정보를 보내기 위해 취할 수 있는 방법은 아무도 범할 수 없는 튼튼한 봉투를 만드는 것이 아니라 누가 건들기 만 하여도 쉽게 부서져 버리는 상태를 이용하는 것이다. 누군가 정보를 얻기 위해서는 전달되는 신호를 측정해야하는데 양자역학에서는 측정 자체 가 원래 신호의 상태를 변형시켜버린다. 따라서 받는 사람은 이 신호가 도청 사실을 결과적으로 알 수 있기 때문에 이를 응용하면 완벽한 암호화 통신을 할 수 있다. 따라서 빛의 양자역학적 특성을 이용한 양자 암호화방법이 제안되었다. 표준적인 BB84 프로토콜의 안전성의 증명은 1996년 D.Mayers에 의해 이루어졌다[38]. 이 증명은 전송 채널과 광자 검출기에 잡음이 존재하며 광원은 완벽할 경우의 양자암호에 대한 것으로 POVM 모델을 사용하였다. 1999년 H.K. Lo 등은 잡음이 존재하는 양자계를 잡음이 없는 양자계로 환원한 후 이를 다시 잡음이 없는 고전계로 바꾸어 안전성을 증명했다[21]. 2000년에는 P. Shor, J.Preskill는 모든 광원과 광검출기의 결점을 도청자의 기저에 무관한 공격에 포함된다고 가정하여 안전성을 증명하였으며, 이 증명은 흔히 Shor-Preskill 증명으로 알려져 있다. 2003년의 Koashi-Preskill 증명에서는 광검출기는 완전하다고 가정하고 광원이 Alice 의 기저정보를 도청자 에게 흘려주지 않지만 안전하지는 않다는 가정을 하고 있다. D. Gottesman, H.K.Lo, N. Lutkenhaus, J. Preskill 등이 2004년 발표한 논문에서는 지금까지의 증명 중 실제 양자암호 시스템과 가장 유사하게 광원과 광검출기가 모두 기저에 대한 약간의 정보를 도청자에게 유출할 경우에 대한 안전성을 증명하였다. 최근까지의 BB84 프로토콜의 안전성 즉 데이터 이동 경로상의 안전성이 입증되더라도 양자암호 시스템 내에서 사용되고 있는 난수열들에 대하여 의사난수를 이용하는 구간을 분석하면 이에 대한 이진병합적 접근으로 부분정보를 유출 원타임패드(One-time pad)의 안정성에 의하여 절대적 도청불가 상태의 암호 구현을 실현한 양자암호 시스템 내에서 사용자의 실난수 사용에 대한 불편함으로 인하여 의사난수를 사용한다. 이는 단일 비트열에서는 입증된 난수성이 다중 비트열의 병합에 의하여 부분정보나 난수성의 오류를 보이는 것을 알 수 있다. 따라서 많은 개발비와 사용상 비용이 많이 드는 문제점을 해결 할 수 있는 실난수를 활용한 양자암호 시스템 개발이 요구되어야 할 것이다[16].

본 논문은 다음과 같이 구성된다.

제 2장에서는 양자암호 시스템을 이해하기 위한 양자역학의 기본적 성질과 관용

암호방식 및 공개키 암호방식을 소개하였다. 또한 NIST의 16가지 난수성 통계 테스트에 대해 다루었고 그 중에서 중요한 난수 테스트인 프리퀀시 테스트, 런 테스트, 시리얼 테스트, 포커 테스트를 자세하게 기술하였다.

제 3장에서는 양자암호 시스템인 BB84 프로토콜, B92 프로토콜, E91프로토콜을 소개하고 이의 벡터 해석적 방법을 기술하였다.

제 4장에서는 BB84 프로토콜 상에서 난수병합에 따른 양자암호 취약성을 제시하고, 실난수를 사용하는 방법으로 문제점을 개선한 광자 발생기를 적용한 의사코드 알고리즘을 제시 적용하였다.

제 5장에서는 양자암호 시스템에서 의사난수 사용에 대한 취약성을 실난수를 사용하는 방법으로 개선한 연구 결과와 향후 활용 분야를 제시하였다.

Ⅱ. 관련연구

A. 양자역학의 기본적 성질

양자암호 시스템을 이해하기위한 기본성질인 불확정성의 원리, 양자중첩, 관측에 의한 사영, 양자얽힘은 다음과 같다.

1. 불확정성의 원리

고전 역학에선 관측하고자 하는 대상의 상태를 전혀 교란시키지 않고 대상에 대한 측정이 가능하다. 하지만 미시세계와 같이 양자역학이 지배하는 세계는 시스템를 교란하지 않고 그 시스템에 대한 정보를 얻을 수 없다. 양자역학에는 서로 상보적인 물리량들이 존재한다. 어떤 입자의 위치를 정확히 측정하면 운동량이 완전히 불확실해지고 반대로 운동량을 정확히 측정하면 입자의 위치가 불확실해진다. 두 측정량 사이에는 항상 불확실성이 존재한다.

2. 양자중첩

양자중첩은 고전정보통신에 있어서의 정보의 기본단위인 비트가 0이나 1중에 어느 쪽이든 한쪽의 값을 반드시 갖는 반면 양자정보통신에 있어서의 양자정보의 기본단위인 큐비트는 동시에 0과 1의 양쪽 값을 가질 수 있는 성질을 말한다. 즉, n개의 큐비트로 $2n$개의 상태를 동시에 표현할 수가 있다. 이 성질을 이용하면 한 번의 데이터 입력만으로 $2n$개의 계산을 동시에 할 수 있다. 이것은 양자컴퓨터의 주요 성질이다.

3. 관측에 의한 사영

관측에 의한 사영이란 양자중첩 상태에 있는 큐비트를 한번이라도 관측하면 동시에 0과 1의 양쪽의 값을 갖고 있었던 상태가 0이나 1의 어느 쪽으로 결정되어 버

리는 성질이다. 즉 중첩의 상태가 해제되고 각각의 비트는 결정된 상태가 되어 버린다. 이 성질을 이용하면 큐비트의 상태변화의 유무에 의해 통신 도중 도청(관측)된지 여부의 판정이 가능하게 된다. 양자암호에 있어서의 양자암호 해독열쇠배포는 이 성질을 활용하고 있다.

4. 양자얽힘

양자얽힘(Quantum Entanglement)은 두 개의 입자가 아무리 멀리 떨어져 있어도 하나처럼 행동한다는 현상이다. 두개의 큐비트는 4개의 기본 양자 상태 {|00〉 ≡ |0〉|0〉 , |0〉|1〉 , |1〉|0〉 ,|1〉|1〉 }의 중첩으로 나타낼 수 있다. 이들 기본 양자 상태들은 두 큐비트의 기본 상태들의 곱으로 나타낼 수 있지만 이들이 중첩된 두 큐비트의 일반적인 상태는 극히 예외적인 경우를 제외하고는 두 큐비트의 곱으로 나타낼 수 없다. 두 큐비트의 곱으로 나타낼 수 있는 극히 예외적인 경우를 분리 가능한(Separable) 상태라고 하고 그렇지 않은 경우를 얽힌(Entangled) 상태라고 한다. 분리 가능한 상태는 한 큐비트를 측정하여 어떤 결과를 얻더라도 다른 큐비트에는 아무런 영향도 미치지 않는다. 하지만 얽힌 상태에서는 한 큐비트를 측정하면 다른 큐비트의 상태에 영향을 미치게 된다.

B. 관용 암호 방식

관용 암호 방식은 암호화와 복호화에 동일한 키를 사용함으로 공통키 암호 방식 또는 암호화와 복호화 과정이 대칭적이어서 대칭 암호 방식이라 한다. 또한 관용 암호 방식은 수 천년 전부터 사용되어 오고 있는 암호 방식으로 평문의 문자를 다른 문자로 치환 또는 문자의 위치를 바꾸는 전치과정으로 구성된다.

1. 치환 암호

a. 시프트 암호

치환 암호(Substitution Cipher)는 평문 문자를 암호문 문자로 일대일 대응시켜 암호화하는 방식이다. 치환 암호 중에 가장 간단한 방법이 영문자를 [그림 2.1]과 같이 나열하고 치환전의 문자를 일정 방향으로 일정 간격 시프트 시키는 방법으로 문자열 “defghia”를 암호화하면 문자열 “GHIJKLD”가 된다[18].

평문	d	e	f	g	h	i	a
암호문	G	H	I	J	K	L	D

[그림 2.1] 시프트 암호

b. Affine 암호

Affine 암호는 더하기와 곱하기를 합쳐놓은 암호라 할 수 있다. 먼저 알파벳 한 글자를 선택해서 그 수에 대응 하는 정수에 어떤 수 m을 더한 후 n을 곱하여 나눈 정수를 26으로 나눈 나머지를 다시 알파벳으로 고치는 암호기법이다. 평문에 키를 곱하여 법연산한 후 평문을 암호문으로 치환하는 $C = M \times K \bmod 26$방식은 암호문을 평문으로 복원하는 복호화 과정은 $M = C \times K^{-1} \bmod 26$으로 용이하게 복호화가 이루어지나 이 암호 방식이 암호문에서 평문으로 유일하게 복원되려면 K와 26은 서로소 즉, $\gcd(K, 26) = 1$이어야 한다. 이 의미를 합동식 $ax \equiv b \bmod m$에서 모든 $b \in Z_m$에 대하여 x가 유일한 해를 가지려면 $\gcd(a, m) = 1$ 이어야 한다는 것이다. 따라서 m과 서로소인 k는

$K = 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25$로 12개의 키가 존재하게 되어 시프트 암호보다 더욱 취약한 암호가 된다[1]. 시프트 암호화 곱셈을 조합한 암호를 고려하면 곱셈 합동식에 또 하나의 키를 삽입한 경우로 식 (2.1)의

$$C = (K_1 M + K_2) \bmod 26 \qquad (2.1)$$

암호 방식을 Affine 암호 방식이라 한다. 위의 함수를 Affine 함수라 하며 앞에서 설명한 단순 시프트 암호는 $K_1 = 1$인 경우를 말하며 이 Affine 암호 방식도 암호문이 유일하게 평문으로 복호화 되기 위해서는 Affine 함수의 역함수가 존재해야 한

다.

$$C \equiv (K_1 M + K_2) \bmod 26 \tag{2.2}$$

$$K_1 M \equiv (C - K_2) \bmod 26 \tag{2.3}$$

Affine 암호 방식의 키 숫자는 $K_1 = 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25$의 12개의 K_1과 26개의 K_2의 조합이 키가 될 수 있다.

c. 치환 암호

평문 각 문자가 다른 문자로 하나씩 대치되는 암호이기 때문에 평문의 단문자 분포가 대응되는 문자의 빈도와 같게 된다. 다표식 치환 암호는 다중 치환 암호와 단순 대치 암호와 동음이의 치환 암호의 문제점을 보완한 암호이다. 이 방식은 다표식 치환표를 2회 이상 적용하는 방식으로 치환표의 갯수를 d라 하면 다표식 치환 암호 방식은 주기 d를 갖는다.

다표식 치환 암호의 암호화 과정을 $E_{Ki}(N) = f_i(N)$이라 하면, 평문 식 (2.4)에

$$M = n_1 n_2 \cdots n_d n_{d+1} n_{d+2} \cdots n_{2d} n_{2d+1} \cdots \tag{2.4}$$

대하여 암호화 과정을 반복 적용하면 식 (2.5)의 암호문은 다음과 같이 생성된다.

$$\begin{aligned} C &= N_{Ki}(N) \\ &= f_1(n_1), f_2(n_2), \ldots, f_d(n_d), f_1(n_{d+1}), f_2(n_{d+2}), \ldots, f_d(n_{2d}), \ldots \end{aligned} \tag{2.5}$$

[표 2.1] Vigenere Cipher 변환표

평문	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	B	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	HI	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	KL	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

다표식 치환 암호 방식으로는 프랑스의 암호학자 Blaise De Vigenere가 고안한 Vigenere 암호가 있다. Vigenere 암호 방식의 암호화 과정은 먼저 d를 결정한 후 다표식 치환표를 적용하는 순서를 정해야 하며, 일반적으로 d를 결정하는 방법은 키 워드(Key Word)를 선택하여 그 키 워드의 길이로 정하고 치환표의 적용 순서는 키 워드의 문자로 정한다. 키 워드로 $cipher$를 선택한 경우 $d=6$이고, 다표식 치환 변환표의 적용 순서는 $cipher$로 결정된다.

(1) Hill 암호

Lester S. Hill에 의해 고안된 암호 방식으로 d를 정수라 하면 암호문은 $(Z_{26})^d$개의 평문 중에 d개의 문자의 선형 결합으로 표시되며 $d=2$일 때 평문을 $M=(m_1\ m_2)$로 암호문을 $C=(c_1\ c_2)$라고 표시하면 c_1과 c_2는 m_1과 m_2의 선형 결합으로 표시된다.
다시 행렬로 표시하면 식 (2.6)과 같다.

$$(c_1\ c_2)=\begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}\begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \tag{2.6}$$

일반적으로 Hill 암호의 키는 $d\times d$행렬 K로 표시되고 K의 i행과 j열의 요소는 K_{ij}로 나타낸다. $d\times d$ 행렬로 구성되는 Hill 암호는 식 (2.7)로 표시된다.

$$(c_1 c_2 \cdots c_d) = \begin{pmatrix} K_{11}\, K_{12} \cdots K_{1d} \\ \vdots \\ K_{d1}\, K_{d2} \cdots K_{dd} \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_d \end{pmatrix} \tag{2.7}$$

즉 $C=KM$이다. Hill 암호 방식은 선형 변환(Linear Transformation)으로 평문으로부터 암호문을 생성한다[22].

복호화 과정은 $C=KM$의 양변에 행렬 K의 역행렬 K^{-1}를 곱하면 가능하다.

$$\begin{aligned} CK^{-1} &= KMK^{-1} \\ &= M \end{aligned} \tag{2.8}$$

선형 대수에서 행렬 $A=(a_{ij})$가 $l\times m$의 행렬이고 행렬 $B=(b_{jk})$가 $m\times n$ 행렬이라면 행렬의 곱 $AB=(c_{ik})$ $(1\leqq i\leqq l,\ 1\leqq k\leqq m)$은 $c_{ik}=\sum_{j=1}^{m} a_{ij}\, b_{jk}$와 같다.
$C=AB$의 i행 K열의 성분은 행렬 A의 i행과 행렬 B의 K열은 차례로 곱한 합이 되며 AB는 $l\times n$행렬이다. 교환법칙이 성립하지 않으므로 I_d는 $d\times d$행렬로 대각선

이 1이고 나머지는 모두 0인 $d \times d$ 행렬을 말한다. 그러나 모든 행렬이 역행렬을 갖지는 않으므로 복호화 과정에서 $C = KM$에 대하여 역행렬 K^{-1}가 존재하면 $CK^{-1} = (KM)K^{-1} = M(KK^{-1}) = MI_d = M$이 존재한다.

(2) Playfair 암호

Playfair 암호는 영국의 C. Wheatstone에 의해 개발되어 그의 동료인 Baron Playfair에 의해 발표된 암호 방식이다. Playfair 암호 방식은 [표 2.2]와 같이 암호화한다.

[표 2.2] Playfair 암호표

T	I	G	E	R
S	A	B	C	D
F	H	K	L	M
N	O	P	Q	U
V	W	X	Y	Z

암호화하기 전 평문을 두 문자씩 나누고 동일한 문자가 연속될 때는 연속되는 동일한 문자 사이에 임의의 문자를 삽입한다. 이러한 과정을 거친 평문의 문자수가 기수이면 임의의 문자하나를 첨가하며 암호화는 평문의 두 문자 m_1, m_2를 암호문 두 문자 c_1, c_2로 변환 암호화 절차는 [표 2.3]과 같다.

[표 2.3] 암호절차

순 서	암 호 절 차
1	평문 m_1과 m_2가 Playfair 암호표의 같은 행에 있으면 c_1, c_2는 각각 m_1, m_2의 우측 문자가 된다.
2	평문 m_1과 m_2가 Playfair 암호표의 같은 열에 있으면 c_1, c_2는 각각 m_1, m_2의 아래 문자가 된다.
3	평문 m_1과 m_2가 Playfair 암호표의 서로 다른 행, 또는 다른 열에 있으면 m_1과 m_2를 절점으로 하는 사각형을 만들어 m_1행의 다른 절점은 c_1, m_2행의 다른 절점은 c_2로 결정한다.

2. 적 암호(Product Cipher)

암호 강도를 향상시키기 위해 전치와 치환을 혼합한 암호 방식으로 그 대표적인 예로 제 1차 세계 대전 때 독일군이 사용하던 ADFGVX 암호가 있으며 관용 암호 방식은 적 암호 방식을 이용한다[1].

a. ADFGVX 암호

ADFGVX의 여섯 개의 문자를 행과 열로 나열한 다음 36개의 열과 행이 직교하는 위치에 26문자와 10개의 숫자를 무작위로 대입하여 암호화 하며 [표 2.4]와 같이 ADFGVX 표에 36개의 문자와 숫자를 직교하는 위치에 무작위로 배치한 다음, 평문을 치환 과정 후 중간 암호화 과정을 거친다. 이 때 36개의 평문 문자에 대한 치환과정은 먼저 행을 선택하고 다음 열을 선택 평문 a에 대한 중간 치환 암호는AF이다.

[표 2.4] ADFGVX 암호 치환표

	A	D	F	G	V	X
A	f	x	a	9	u	1
D	n	g	0	1	d	o
F	5	b	k	2	h	z
G	m	j	s	y	t	v
V	7	4	3	e	8	i
X	c	w	q	6	r	p

중간 암호화된 치환 결과를 연속적으로 나열한 다음 전치 키 워드에 따라 중간 암호화된 결과를 나열하면 최종 암호화 결과를 얻을 수 있다.

3. 스트림 암호

스트림 암호는 이진화 된 평문 스트림과 이진 키 스트림의 배타적 논리합 연산으로 암호문을 생성한다. 스트림 암호(Stream Cipher)는 대칭 암호화 알고리즘의 일종으로 어떠한 블록 암호보다도 빠르다. 평문을 일정 블록으로 나누어 각 블록을 동일한 키로 암호화 하고 평문 블록 m_i를 암호화 키 K로 암호문 C로 암호화 한다. 블록 암호는 식 (2.9), 식 (2.10)으로 표현된다.

평문 $$M = m_1 m_2 m_3 \ldots \tag{2.9}$$

암호문 $$C = E_K(m_1) E_K(m_2) E_K(m_3) \ldots \tag{2.10}$$

또 다른 스트림 암호(Stream Cipher)방식은 평문에 연속되는 키 계열(Key Stream)을 적용시켜 암호화 한다.
평문 $M = m_1 m_2 \ldots$ 에 키 계열 $Z = z_1 z_2 \ldots$ 를 적용한 암호는 식 (2.11), 식 (2.12)로 표현된다.

평문 $M = m_1 m_2 m_3 \ldots$ (2.11)

암호문 $C = E_{z_1}(m_1) E_{z_2}(m_2) E_{z_3}(m_3) \ldots$ (2.12)

스트림 암호에서 가장 중요한 역할을 하는 것이 키 계열 발생이다. 특별한 함수를 이용 키 K와 평문으로 키 계열을 발생시키는 방법이 있다.

$$z_i = f(K, m_1, m_2, \ldots, m_{i-1}) \tag{2.13}$$

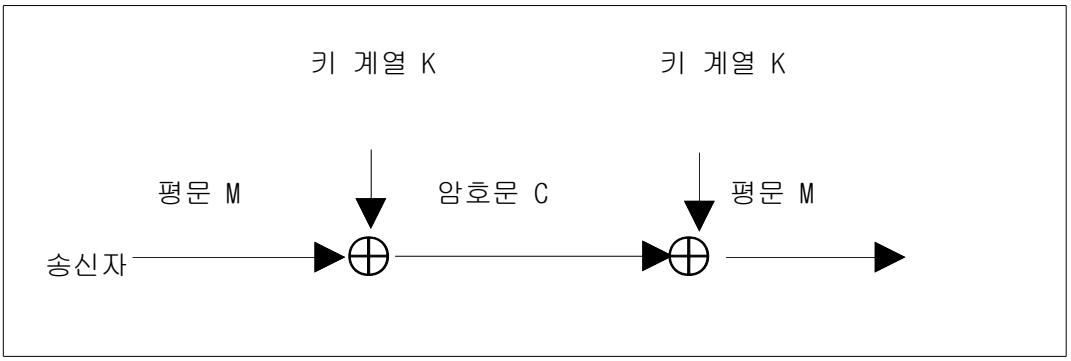
발생된 키 계열 식 (2.13)은 평문 m_i의 암호화 $E_{z_i}(m_i)$에 사용하며
평문 $M = m_1 m_2 m_3 \ldots$ 을 암호화 할 때는 식 (2.14)의 순서로 계산하며

$$z_1 C_1 \; z_2 \; C_2 z_3 \; C_3 \; \ldots \tag{2.14}$$

암호문 $C_1 \; C_2 \; C_3 \ldots$ 의 복호화 과정은 식 (2.15)의 순서로 계산한다.

$$z_1 \; m_1 z_2 \; m_2 z_3 \; m_3 \; \ldots \tag{2.15}$$

블록 암호 방식은 스트림 암호 방식의 특수한 경우로 $i \geq 1$에 대하여 $z_i = K$의 경우라고 생각할 수 있으며 키 계열이 평문 문자와 무관한, K만의 함수인 경우 이런 스트림 암호를 동기 스트림 암호라 한다. 키 계열 z가 K의 영향을 받아 K를 시드(Seed)라 하며 동기 스트림 암호에서 $i \geq 1$에 대하여 $z_{i+d} = z_i$일 때 주기적인 동기 스트림 암호 방식이 되며 주기가 d가 된다. 스트림 암호 방식의 구성 방식은 [그림 2.2]와 같다.



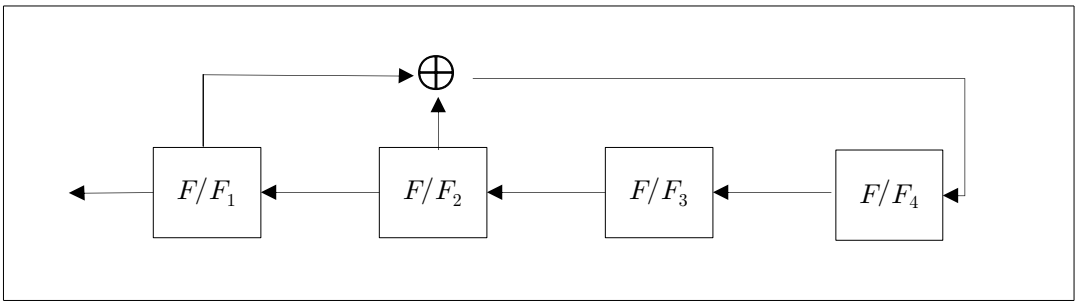
[그림 2.2] 스트림 암호

스트림 암호 방식의 암호 강도는 키 계열의 무작위성이 결정하며 키 계열 발생 방법으로는 동전을 던져서 나타나는 양면을 0과 1로 배정하여 키 계열을 만드는 방법이다. 대부분 일정 주기의 키 계열 발생 방법을 이용한다. 우선 $k_1 k_2 \ldots k_d$에 대하여 $1 \leq i \leq d$에서 $z_i = k_i$가 되도록 만들고 식 (2.16)과 같이 이단 선형 순환 관계를 이용하여 키 계열을 계속 생성시킨다.

$$z_i + d = \sum_{j=0}^{d-1} b_j z_{i+j} \mod 2 \tag{2.16}$$

단, 상수 $b_0, b_1, \ldots, b_{d-1} \in Z_2$는 사전에 정해진 정수이다.

순환은 각각 기간이 그 이전 d기간에 좌우되고, 선형 함수이므로 선형 순환이라 한다. 순환이 d가 되기 위해서는 반드시 $b_0 = 1$이어야 한다. 또 키 계열 K의 주기가 d가 되도록 $b_0, b_1, \ldots, b_{d-1}$을 선택하여야 하며 $K = k_0 k_1 k_2 \ldots k_d$의 값이 모두 0이 되지 않도록 해야 한다. 만일 키 K가 모두 0이 되면 키 계열이 모두 0이 되어 암호문과 평문이 동일해져 암호 강도에 문제가 발생한다. 일반적으로 키 계열은 선형 궤환 시프트 레지스터(Linear Feedback Shift Register, LFSR)를 이용하여 생성한다. d단 시프트 레지스터에 각 플립플롭의 출력을 선형 회로에 입력시켜 그 출력을 키 계열로 사용한다.



[그림 2.3] 4단 선형 궤환 시프트 레지스터

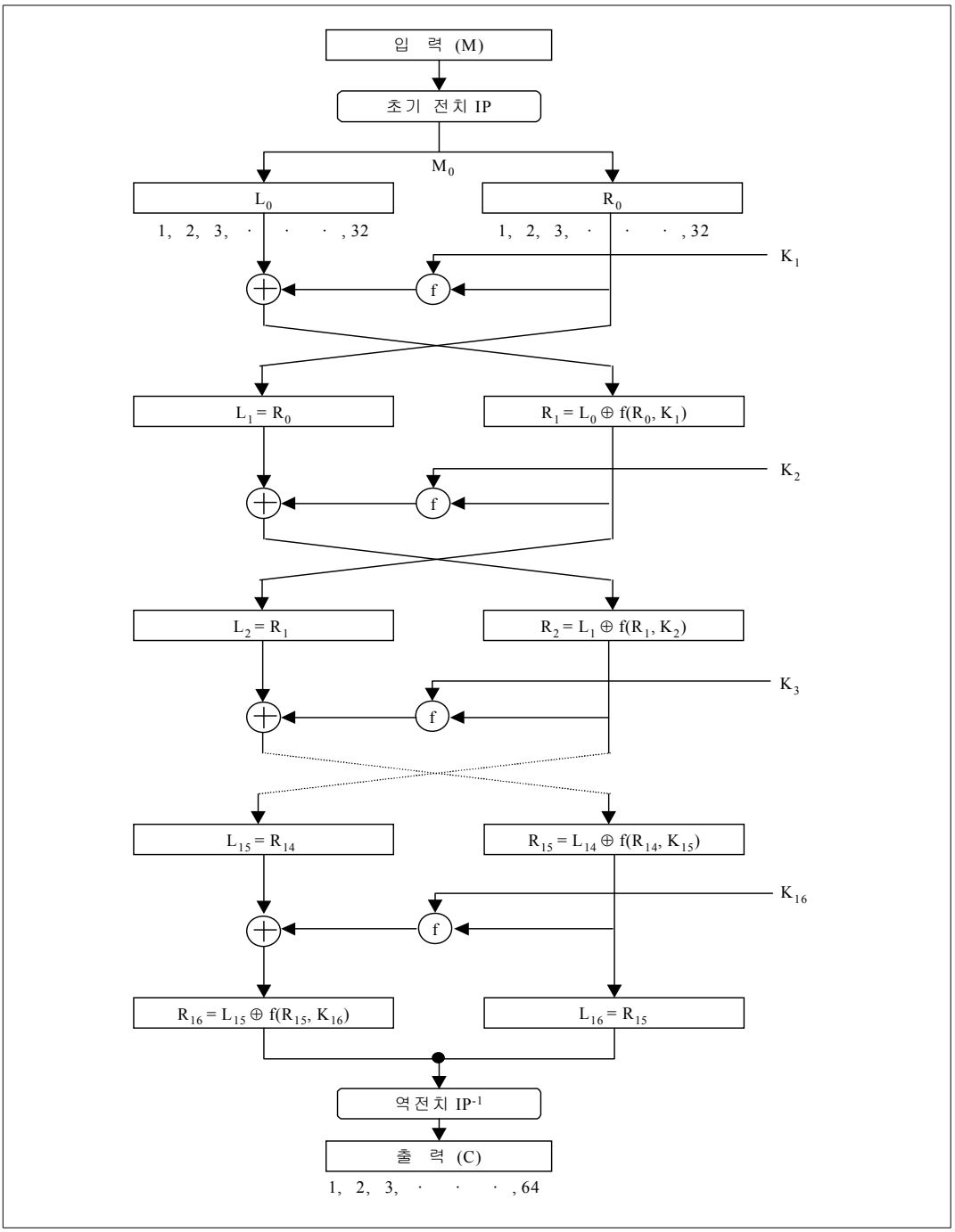
시프트 레지스터를 구성하고 있는 플립플롭 F/F_i의 초기값은 모두 0이어서는 안된다. 만일 초기값이 모두 0이면 선형 궤환 시프트 레지스터의 출력 키 계열은 계속해서 0만 출력된다. 이 경우 선형 궤환 시프트 레지스터를 이용한 스트림 암호는 암호문과 평문이 동일해진다.

4. DES

블록 암호 알고리즘의 대명사인 DES는 1972년 미국 상무성의 NBS에서 컴퓨터와 통신 정보를 보호할 목적으로 표준 암호 알고리즘을 공모하고 IBM이 개발한 알고리즘 이다. 표준 암호 알고리즘은 DES(Data Encryption Standard)로 약칭화 되었으며 상무성은 DES의 안정성을 5년마다 검토하기로 하고 1993년 검토에서 1998년까지는 DES를 표준 암호 알고리즘으로 사용키로 하고 그 이후에는 새로운 표준으로 대체할 것을 검토하기로 하였다. DES 암호 알고리즘 지정은 순수 민간인용으로만 사용하고 있으며 ANSI(American National Standards Institute)에서도 표준으로 지정하여 금융 기관에서 많이 사용하고 있다[1,26,28].

a. 암호화 과정

DES의 암호화 과정은 [그림 2.4]와 같으며



[그림 2.4] DES 암호화 과정

DES는 64비트 한 블록에 대해 먼저 IP치환을 해주고 그 결과물을 32씩 L_0와 R_0로 나누고 암호문 64비트로 변환시키는 암호 방식으로 64비트 키를 사용한다. 암호화 과정에는 54비트만이 적용된다. 암호 알고리즘의 기본 동작은 전치, 치환, mod 2 연산으로 구성되어 있으며 치환은 S-Box라는 치환 장치에서 이루어지나 전치, 치환, 그리고 mod 2연산으로 구성된 동작 과정은 다음과 같이 크게 세 가지 과정으로 나눌 수 있다.

첫째, 평문 M의 64비트는 초기 전치(Initial Permutation) IP를 거쳐 $IP(M) = M_0$는 32비트씩 나누어져 L_0, R_0로 나누어진다. 초기 전치 IP는 평형 전치로 [표 2.5]와 같다. 표의 숫자는 평문 64비트의 위치를 나타낸다. 즉 58번째 비트를 1번째 비트로 50번째 비트를 2번째 비트로 64비트의 위치를 변경시킨다.

둘째, 초기 전치 출력 R_0, L_0는 식 (2.17)과 식 (2.18)과 같은 함수 계산을 16회 반복하며

$$L_i = R_{i-1} \tag{2.17}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \tag{2.18}$$

L_i, R_i는 32비트씩의 중간 데이터를 말한다. 서브 키 K_i는 48비트의 DES 암호화 키로써 $K_1, K_2, \ldots, K_{16}$의 값은 서로 다르며 16회 암호화 과정에 사용되며 함수 f는 S-Box를 포함한 치환 과정을 의미한다.

[표 2.5] 초기 전치 IP

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

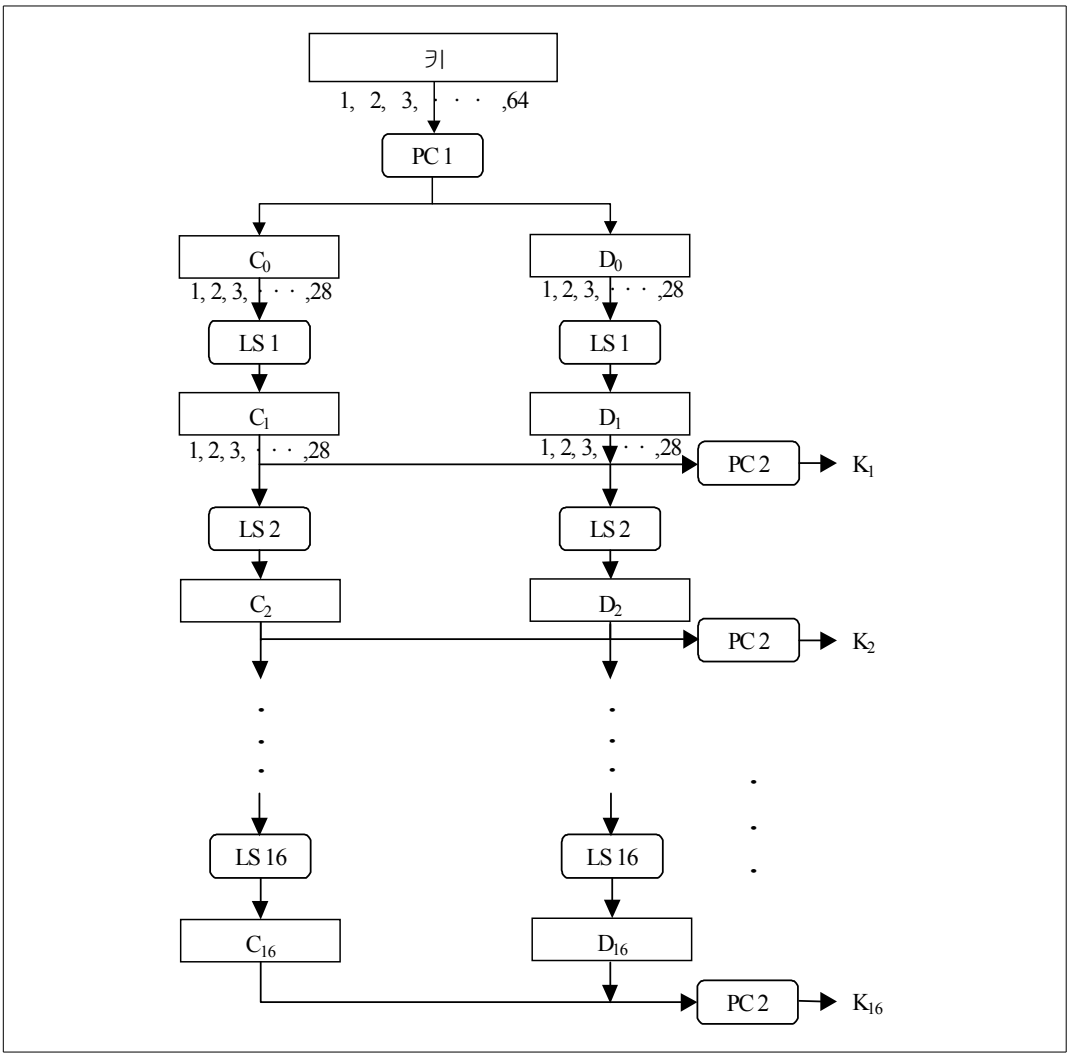
셋째, R_{16}과 L_{16}은 초기 전치의 역전치인 IP^{-1}를 거쳐 64비트의 암호문이 된다. 즉 $C = IP^{-1}(R_{16}, L_{16})$으로 L_{16}과 R_{16}이 서로 반대로 되어 있다. IP^{-1}는 IP의 역전치로 [표 2.6]와 같다. IP에서 58번째 비트가 1번째 비트로 전치되었기 때문에 반대로 IP^{-1}에서는 1번째 비트가 58번째 비트로 전치되어야 하며 1이 58번째인 마지막 행 두 번째 열에 위치하는 것을 알 수 있으며 IP에서 50번째 비트가 2번째 비트로 전치되었기 때문에 IP^{-1}에서는 2번째 비트가 50번째 비트로 전치된다.

[표 2.6] 초기 전치의 역전치 IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

b. 키 생성 과정

DES 암호는 서로 약속한 키로 K로부터 16개의 서브키 로 이루어진 키 스케쥴러 $K_1, K_2, \cdots, K_{16}$을 만들어서 암호화에 사용한다. 생성 과정을 살펴보면 키 K가 키 스케쥴러에 입력되면 16개의 서브키 $K_1, K_2, \cdots, K_{16}$이 출력되며 K의 길이는 64비트 열로 8번째 비트마다 패리티 검사용 비트가 포함되어 있어 실제의 키 길이는 56비트이다. 키가 키 스케쥴러에 입력되면 8, 16, 24, 32, 40, 48, 56, 64번째 기수 패리티 비트를 무시한 채 키 스케쥴러가 동작한다. 키 스케쥴러의 구성은 [그림 2.5]와 같다.



[그림 2.5] 키 스케줄러

첫째, 64비트의 키로부터 패리티 검사 비트를 제거한 나머지 56비트를 전치 $PC-1$(Permuted Choice 1)에 따라 전치를 시킨 후 $PC-1(K)=C_{0,}D_0$로 나눈다. C_0와 D_0는 28비트씩 나누어진다. $PC-1$의 전치는 [표 2.7]과 같다.

[표 2.7] 키 전치 $PC-1$

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

둘째, 식 (2.19)의 C_i와 식 (2.20)의 D_i는 각각 LS_i에서 왼쪽으로 Cyclic Shift된 후 전치 $PC-2$(Permuted Choice 2)에 따라 56비트가 48비트로 축약 전치된다.

$$C_i = LS_i(C_{i-1}) \tag{2.19}$$

$$D_i = LS_i(D_{i-1}) \tag{2.20}$$

$$K_i = PC-2(C_i D_i) \tag{2.21}$$

이 때 $PC-2$는 축약 전치로 [표 2.8]과 같으며, Cyclic Shift 회수는 LS의 위치에 따라 [표 2.9]와 같이 수행된다.

[표 2.8] 축약 전치 $PC-2$

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	17	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

[표 2.9] 키 스케쥴러 LS의 Shift 수

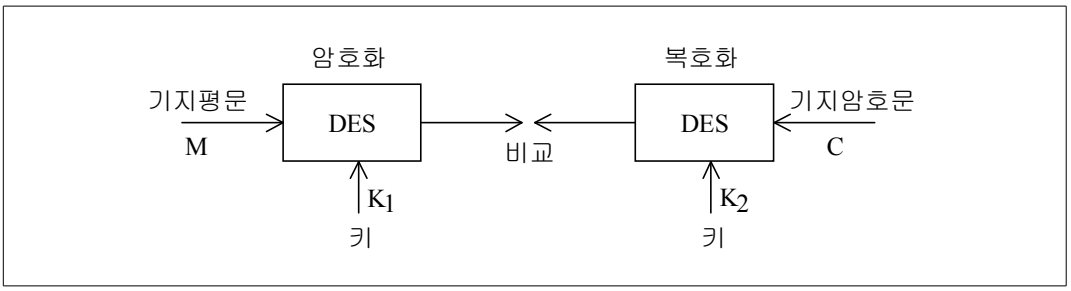
위치	시프트	위치	시프트
LS_1	1	LS_9	1
LS_2	1	LS_{10}	2
LS_3	2	LS_{11}	2
LS_4	2	LS_{12}	2
LS_5	2	LS_{13}	2
LS_6	2	LS_{14}	2
LS_7	2	LS_{15}	2
LS_8	2	LS_{16}	1

c. 2중 DES 암호 방식의 사용

DES 알고리즘 자체는 변형시키지 않고 DES의 안전성을 증대시키기 위한 방법 중의 하나로 2개의 다른 키로 2번 암호화를 수행하는 식 (2.22)의 2중 DES는 외형상으로는 2배의 키 길이지만 메모리 용량이 충분하면 57비트 효과밖에 얻지 못한다. [그림 2.6]과 같이 기지 평문 공격에서 DES의 반복 적용한 경우의 소모적 공격

을 생각해보자. 먼저 기지 평문을 모든 키로 암호화하여 얻은 암호문과 기지 암호문을 모든 키로 복호화 한 평문을 비교하면 두 개의 암호 키를 찾을 수 있으므로 실제로는 키의 길이가 한 비트 더 늘어난 효과밖에 얻지 못한다.

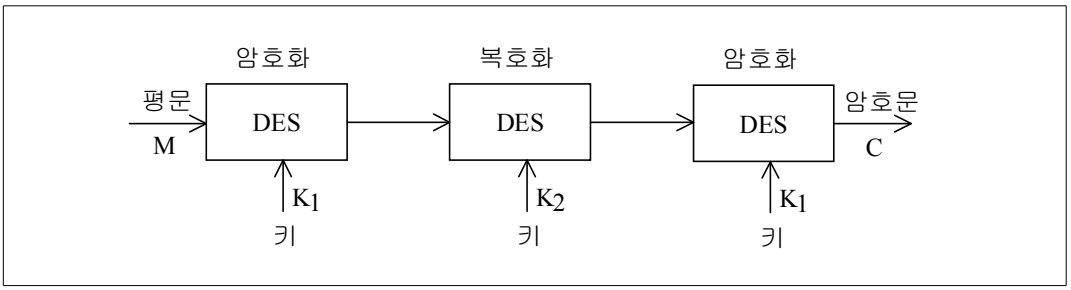
$$E_{K_2}(E_{K_1}(M)) = C \tag{2.22}$$



[그림 2.6] 2중 DES 암호 방식의 소모적 공격

그러나 [그림 2.7]과 같이 두 개의 암호 키를 사용하여 첫 번째 키로 암호화하고 다시 두 번째 키로 복호화 한 다음 또 다시 첫 번째 키로 암호화하면 식 (2.23)과 같은 강한 암호를 얻을 수 있다.

$$E_{K_1}(D_{K_2}(E_{K_1}(M))) = C \tag{2.23}$$



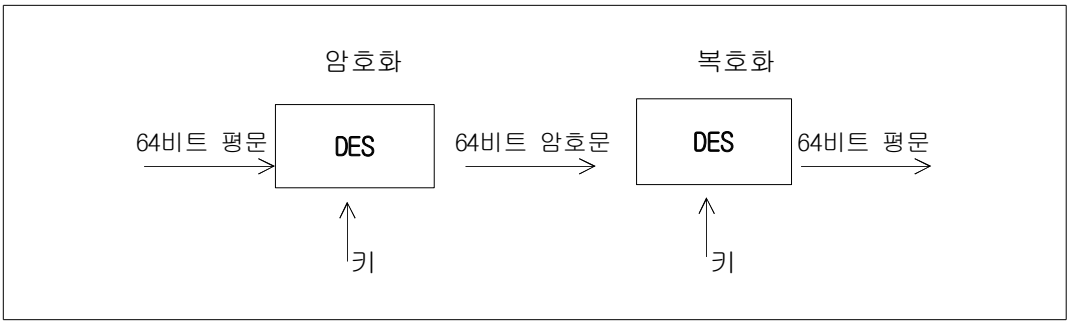
[그림 2.7] 3중 DES 암호 방식

5. DES 암호 방식의 운용 모드

DES 암호 방식의 이용 모드는 ECB(Electronic Code Book)모드, CBC(Cipher Block Chaining)모드, CFB(Cipher Feed Back)모드, 그리고 OFB(Output Feed Back)모드 등으로 동작시킨다.

a. ECB(Electronic Code Book) 모드

ECB 모드는 DES 암호 방식의 사용 방식 중 가장 간단한 방식으로 평문을 64비트씩 나누어 암호화하는 방식이며 동일한 평문 블록 모양에 따라 항상 동일한 암호문이 출력되므로 해독 가능성을 높게 만들며 DES 암호 방식의 키 암호화에 사용된다[1].



[그림 2.8] EBC 모드 동작

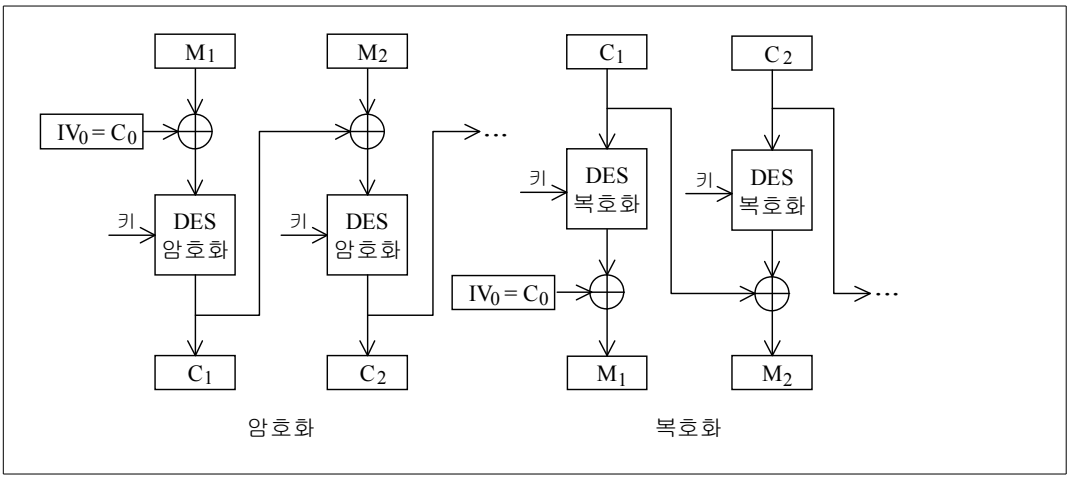
b. CBC(Cipher Block Chaining) 모드

CBC 모드는 출력 암호문이 다음 평문 블록에 영향을 미쳐 각 암호문 블록이 전단의 암호문의 영향을 받도록 만든 방식으로 ECB에서 발생하는 동일한 평문에 의한 동일한 암호문이 발생하지 않도록 구성한 동작 모드이다. CBC 모드 동작은 [그림 2.9]와 같이 입력된 평문 블록 M_1은 초기 벡터 IV_0(Initial Vector)와 배타적 논리합이 되어 DES 암호기에 입력된다. 암호기 출력 암호문 C_1은 다음 단 평문 블록 M_2와 배타적 논리합이 되어 DES 암호기에 입력되며, 식 (2.24), 식 (2.25)와 같이 암호화, 복호화가 되며 CBC 모드는 [그림 2.9]와 같다.

$$\text{암호화} \quad C_i = E_K(M_i \oplus C_{i-1}) \tag{2.24}$$

$$\text{복호화} \quad M_i = D_K(C_i) \oplus C_{i-1} \tag{2.25}$$

$$\text{단,} \quad C_0 = IV_0$$



[그림 2.9] CBC 모드 동작

c. CFB(Cipher Feed Back) 모드

CFB 모드도 CBC 모드와 유사하지만 평문 블록이 동일한 경우 동일한 암호문이 나타나지 않도록 전단의 암호문이 다음 단의 평문에 영향을 미치게 구성하는 방식이다.

암호화, 복호화는 식 (2.26), 식 (2.27)과 같다.

$$\text{암호화} \quad C_i = E_K(C_{i-1}) \oplus M_i \tag{2.26}$$

$$\text{복호화} \quad M_i = E_K(C_{i-1}) \oplus C_i \tag{2.27}$$

$$\text{단,} \quad C_0 = IV_0$$

CFB 모드는 한 비트의 오류가 발생하면 모든 암호문에 영향을 미치게 되며 복호화 과정에서도 암호문 블록 내에 한 비트의 오류가 발생 복호화 된 모든 평문 블록에 영향을 미치게 된다.

d. OFB(Output Feed Back) 모드

OFB 모드는 암호 알고리즘의 출력 블록에서 k비트를 선택한 후, k비트 평문과 배타적 논리합을 하여 암호문을 발생한다. 이 방식에서는 에러 전파 현상이 발생하지 않으며 블록 암호를 사용하여, 블록 간의 의존 관계를 갖는 비트열 암호 이용 모드로 블록 암호의 출력 전부를 입력 레지스터 갱신에 사용하는 비트열 암호화 방법. 송신 측에서는 입력 레지스터에 기반을 두고 블록 암호화하여 그 출력의 비트열과 평문 비트열의 배타적 논리합을 취하여 생성한다. 수신 측에서는 송신 측의 암호화 처리와 같은 방법의 역순으로 복호한다.

C. 공개키 암호 방식

1976년 Diffie와 Hellman은 키를 두 개로 나누어 하나는 암호화 키, 다른 하나는 복호화 키로 사용한다. 암호화 키는 공개 목록에 등록 공개하고 복호화 키는 개인이 비밀리에 보관한다. 그러므로 암호화 키는 공개키, 복호화 키는 비밀키 라 부른다. 공개키 암호 방식은 암호화와 복호화를 분리하여 서로 다른 키를 사용하고 있어 비대칭 암호 방식(Asymmetric Cryptosystem)이라 부르기도 하고 키를 2개 사용한다고 해서 Two Key 암호 방식이라고도 한다. 관용 암호 방식은 암호화와 복호화가 한 개의 키로 이루어지고 있으므로 공통키 암호 방식 혹은 대칭 암호 방식이라고 한다. 공개키 암호 방식은 소인수 분해와 이산대수 문제, 배낭 문제 등을 이용하여 실현하고 있다.

1. RSA 암호 방식

비밀키 암호법에 있어서 가장 큰 문제점은 바로 메세지의 전달자와 수신자가 똑같은 비밀키를 다른 사람들이 모르게 공유해야 한다는 점이다. 비밀키 암호 기법이 갖는 위와 같은 키 관리의 문제를 해결하기 위해서 제안된 것이 바로 공개키 암호 기법이다. 1976년 공개키 암호 방식의 개념을 발표한 후, 1978년 MIT의 Rivest, Shamir와 Adleman이 처음 RSA라고 하는 공개키 암호 방식을 제안하였다. 이들은 합성수 의 소인수 분해의 어려움을 이용 RSA 암호 방식을 실현한다[7].

a. 안전성

암호 안전성은 소수 p와 q에 달려있다. 공개 암호화 키 K_e와 n을 가지고 간단하게 비밀 복호화 키 K_d를 찾을 수 있다면 RSA 암호 방식은 쉽게 해독된다. 또한 n으로부터 소수 p와 q를 찾을 수 있다면 즉, n의 소인수 분해가 가능하면 Euler 함수 $\phi(n)$을 찾게 되어 유클리드 호제법으로 공개 암호화 키 K_e로부터 비밀 복호화 키 K_d를 간단하게 찾아낼 수 있다.

RSA 암호 방식의 안전성을 보장받기 위한 소수 p와 q의 선택 조건과 공개 암호

화 키 K_e와 비밀 복호화 키 K_d의 조건들이 부가적으로 필요하다. 소수 p와 q는 p와 q는 거의 같은 크기의 소수이며 $p-1$과 $q-1$은 큰 소수를 인수로 가져야 하며 $p-1$과 $q-1$의 최대 공약수는 작아야 RSA 암호 방식이 안전하다.

b. 이산 대수 문제

이산 대수 문제란 큰 소수 p로 만들어진 집합 Z_p상에서의 원시 원소를 g라 할 때 $g^x \equiv y \bmod p$의 g와 y값을 알고 있어도 $\log_g y \equiv x$를 구하는 것이 어렵다는 점이다.

D. 난수성 테스트

초기값을 이용하여 이미 결정되어 있는 의사난수 생성기에 의해 생성되는 난수는 생성 방법이 결정되어 있지 않으며, 다음에 생성될 값을 전혀 예측할 수 없으나 의사 난수 생성기에 의해 생성되는 수는 초기값을 알면 계산될 수도 있으므로 실 난수와 구별하여 의사난수라 부른다. 좋은 의사난수 생성기는 입력과 출력을 통한 공격을 막기 위한 방법으로 중간 상태변수를 알 수 없게 하여야 한다. 이를 위해서는 초기값과 상태값들을 일정 출력과 일정 시간이 흐른 후에는 갱신과정을 통하여 새롭게 입력되고 변화하여야한다. 미 NIST에서 제공하는 의사난수 테스트 16가지 난수성 검정에서 사용한 통계적 검정에 대해 다루었고 그 중에서 중요한 난수 테스트인 프리퀀시 테스트, 런 테스트, 시리얼 테스트, 포커 테스트를 자세하게 기술한다 [6,10].

1. NIST 의사난수 테스트

a. Frequency test

주어진 수열 $a_1, a_2, \cdots, a_n$을 길이 n인 비트열이라고 하자. Frequency test는 수열이 난수성을 만족하는 경우, 각 비트에서 0과 1이 나타날 확률이 $\frac{1}{2}$이라는 가정에

근거한 검정방법이다[17].

b. Frequency test within a block

Frequency test와 같은 개념이지만 0과 1의 수를 전체 비트열에서 조사하는 것이 아니라 길이가 M인 부분 비트열에서 조사하는 것이다. $a_1, a_2, \cdots, a_n$을 길이 n인 비트열이라고 하고 각 부분 비트열을 $(a_1, a_2, \cdots, a_M) \cdots, (a_{n-M+1}, a_{n-M+2}, \cdots, a_n)$라 하자. 이때, i번째 블록에서 1의 개수를 N_i라 하면 $\sum_{i=1}^{N} N_i = n_i$이다. 각 블록에서의 1의 개수가 근사적으로 $\frac{M}{2}$이 되는지를 조사하는 검정방법이다.

c. Runs test

길이가 n인 비트열 $a_1, a_2, \cdots, a_n$에서 0에서 1로 혹은 1에서 0으로 변화(Change)하는 정도가 추정치에 근접한지를 확인하는 검정방법이다. 비트열이 랜덤하다고 가정하면 변화가 일어날 확률은 각 비트에서 $\frac{1}{2}$이다. 주어진 비트열이 난수열이라면 전체 비트열에서 1의 수 N_i의 비율이 $\frac{n_i}{n}$이 $\frac{1}{2}$이어야 한다. 이는 Frequency test에서 검정한 방법이다.

d. Test for the longest run of ones in a block

길이가 n인 비트열 $a_1, a_2, \cdots, a_n$을 N개의 M비트 블록으로 분할하여 각 블록에서의 1의 연속되는 최대 런(runs)의 수를 고려해보자. Runs test에서는 1이 연속되는 최대 런(runs)의 수를 고려해보자. Runs test에서는 전체 비트열에서의 Runs(blocks+gaps)의 수가 평균 $\frac{(n-1)}{2}$에 근접하는가를 검정하는 것이고 Test for the longest runs of ones in a block은 각 블록에서의 1이 연속되는 최대길이의 런의 수가 N개의 블록들에서 균등하게 나타나는가를 알아보기 위한 검정이다. 단, 비

트열이 랜덤하다고 가정하면 최대 길이 런의 정도가 추정치에 근접하여야 한다.

e. Binary matrix rank test

길이가 n인 비트열 $a_1, a_2, \cdots, a_n$을 m비트로 분할하여 $(a_1, a_2, \cdots, a_m) \cdots, (a_{n-m+1}, a_{n-m+2}, \cdots, a_n)$라 하면, Binary rank test는 각 부분 비트열을 m차 벡터로 고려하여 벡터들의 선형종속성을 조사하는 검정방법이다.

f. Discrete fourier transform(Spectral) test

길이가 n인 비트열 $a_1, a_2, \cdots, a_n$ 을 -1, 1로 이루어진 수열 $c_1, c_2, \cdots, c_n$로 다음과 같이 변환하여 $c_1 = 2a_1 - 1, c_2 = 2a_2 - 1, \cdots, c_n = 2a_n - 1$이 이렇게 변환한 길이가 n, 주기가 m인 수열 $(c_1, c_2, \cdots, c_n)$을 $n = 2^k$차 벡터공간에서 고려하면 Spectral test는 $n = 2^k$ 차 벡터공간에서 주어진 수열을 벡터로 표현하였을 때 벡터의 길이가 얼마나 균일하게 분포되어 있는가를 조사하는 검정방법이다.

g. Non-overlapping template matching test

m비트 Template을 길이가 M인 비트열에서 고려하면 Non-overlapping matching test는 주기성이 없는 패턴(Pattern)의 Template들의 빈도를 조사하는 검정방법이다.

h. Overlapping template matching test

m비트 Template을 길이가 M인 비트열에서 고려하면 Overlapping matching test는 비트열 $a_1, a_2, \cdots, a_n$을 M비트 단위로 나누었을 때, $\lfloor n/M \rfloor = N$ 개의 $(a_{kM+1}, a_{kM+2}, \cdots, a_{kM+M})$들에서, $k = 0, 1, 2, \cdots, M-1, 1$이 연속으로 m개인 런의 빈도가 균등하게 분포되어 있는지를 검정하는 방법이다. M비트 블록에서 m비트 Template의 빈도를 조사하는 검정방법이다.

i. Mauer's test

길이가 n인 비트열 $a_1, a_2, \cdots, a_n$에서, 주어진 비트열의 정보손실 없이 압축이 잘 될 수 있는지를 알아보는 검사이며 압축성이 지나친 비트열은 난수성질을 만족하지 못한다고 할 수 있다. 이 검정은 주어진 비트열의 압축률을 계산하는 것이 아니라 압축된 비트열의 길이와 관련된 수치로 비교하는 검정이다. 본 검정은 Frequency, Runs test 등과 같은 일반적인 난수검정 방법들에서 난수성질을 만족하지 못하는 비트열을 탐지할 수 있다는 장점과 표본수가 커야하는 단점이 있다.

j. Lempel-ziv compression test

주로 데이터 압축(Data Compression) 등에 많이 이용되어지고 있는 방법이다. 주어진 비트열에서 각 부분 비트열의 패턴을 조사하는 검정으로서 Information Theory 관점에서 Approximate Entropy, Mauer's test와 유사하다. 이 검정은 비트열의 반복성의 정도를 나타내는 측도이며, 비트열에서 어떤 한 개의 부분 비트열도 이전에는 일어나지 않도록 분할되어진 부분 비트열의 개수, 즉 비트열을 따라 움직일 때 나타나는 새로운 패턴의 개수가 기대 값에 근사한지를 조사하는 검정방법이다.

k. Linear complexity test

전체 길이가 n인 비트열을 부분 비트열로 분할하여 각 부분 비트열의 선형복잡도가 균일하게 분포되어 있는가를 검정한다. 따라서 전체길이가 n인 비트열을 길이가 M인 부분 비트열로 분할하여 각 부분 비트열에서의 선형복잡도의 분포가 평균과 일치하는지를 조사하는 검정방법이다.

l. Serial test

길이기 n인 비트열 $a_1, a_2, \cdots, a_n$에서 Serial test는 길이가 m인 부분 비트열을 겹치게 생성함으로서 2^m개의 부분비트열에 대한 현태의 균일성을 조사하는 검정방법

이다.

m. Approximate entropy test

의사난수에 대한 Entropy test는 기존의 통계적 관전의 검정에 비해 좀 더 일반적인 통계적 모델을 기반으로 한다. Entropy test는 기존에 사용되는 통계적 검정(즉, Frequency test, Poker test, Runs test, Autocorrelation test 등을 포함하는) 이외에 다음과 같은 두 가지 주된 이점을 제공한다. 첫째로, 특정한 유형의 통계적 결점을 찾는 기존의 통계적 검정들과는 달리, Entropy test는 의사난수 발생기들이 가질 수 있는 발생 가능한 결점들에 관한 Class는 유한 메모리를 가지는 Ergodic Stationary Source에 의해 모델화 될 수 이는데, 이는 의사난수 비트 발생기의 실제적인 구현에서 발생 가능한 결점들을 포함하도록 적절하게 나타내 질 수 있다. 둘째로, Entropy test는 의사난수 발생기의 결점에 관한 실제적인 암호학적 중요성을 측정한다. 좀 더 정확하게 말하자면, 검정 매개변수는 키 소스의 비트 당 Entropy를 측정하는데, 이는 공격자가 비밀 키 소스의 통계적 결점에 관한 지식을 부당하게 이용할 때 공격자가 수행하는 최적 키 탐색 전략의 실행 시간과 관련이 있다. 즉, 비밀 키 소스의 비트 당 Entropy는 암호 시스템을 깨기 위해 전 탐색보다 더 빠른 방법이 존재하지 않는다는 가정 하에서 암호 시스템이 효율적인 키 키기를 측정한다. 이러한 두 가지 이점은 유한 메모리 $M \leq L$ (L : 검정 파라미터)을 갖는 이진 Ergodic Stationary Source의 일반화된 Class와 소스의 비트 당 Entropy H와 밀접하게 관련된 통계량 T를 결과로 가지는 조건부 확률 모델을 설정했다는 사실에 기인한다.

n. Cumulative sums test

Cumulative sums test는 ± 1로 구성된 비트열에서 분합의 절대값의 최대값을 가지고서 모델화한 것으로 난수성 조사에 활용되고 길이가 인 수열에서 0을 -1로 변환한다. 그 변환된 수열의 부분 합의 절대값이 크다면 주어진 수열의 부분 수열에서는 0혹은 1인 수가 비슷하다는 의미를 갖는다.

o. Random excursions test

Random walk test는 어떤 지정된 상태(State)에 도달(Visit)하는 수의 분포를 가지고 조사하는 검정이다. 동물의 이동 상태를 가지고서 모델화한 것으로 난수성 조사에 활용 된다. Random Variables x_i는 -1과 1둘 중의 하나의 수이고 $x_1, x_2, \cdots, x_k$을 길이가 k인 비트열이라 하고 부분 합을 $S_k = x_1, x_2, \cdots, x_k$라 하면 여기서 x_i가 1일 확률을 p, -1일 확률을 $q=(1-p)$라 하면 x_i는 -1과 1둘 중의 하나의 수이기 때문에 부분 합 $S_k = 0$이 되는 k즉 원점으로 되돌아오는 상태에 대한 검정방법이다.

p. Random excursions variant test

Random excursions variant test에서는 부분 합 $S_k = x_1 + x_2 + \cdots + x_k$에서 $S_k = 0$이 되는 k즉 원점으로 되돌아오는 수가 균일한가를 검정하는 것이며 Random excursions variant test는 $S_k = 0$이 되는 동안에 $x \neq 0$에 얼마나 되돌아오는 지를 조사하는 것이다.

2. 프리퀀시(Frequency) 테스트

이진 수열 (s_t)의 N개의 항$s_0, s_1, \cdots, s_{t+2}, \cdots, s_{N-1}$을 처음부터 차례로 m개씩 나누면, 다음과 같이 m차원 벡터공간 $GF(2)^n$에 속하는 $n = [\frac{N}{m}]$개의 벡터가 생긴다.

$$
\begin{aligned}
&(s_0, s_1, \cdots, s_{t+2}, \cdots, s_{m-1}) \\
&(s_m, s_{m+1}, \cdots, s_{m+2}, \cdots, s_{2m-1}) \\
&\cdots\cdots\cdots\cdots\cdots\cdots \\
&(s_{(n-1)m}, s_{(n-1)m+1}, \cdots, s_{nm})
\end{aligned}
\tag{2.28}
$$

이제 체 $GF(2) = \{0, 1\}$위의 m차원 벡터공간

$$GF(2)^n = \{(x_1, x_2, \cdots, x_m) \mid x_1, x_2, \cdots, x_m \in GF(2)\} \tag{2.29}$$

에 속하는 2^m개의 벡터를 적당히 번호를 정하여 식 (2.28)에 있는 벡터 중에서

$$\begin{aligned} &(0, 0, \cdots, 0)\text{인 것의 개수를 } n(0) \\ &(0, 0, \cdots, 1)\text{인 것의 개수를 } n(1) \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ &(1, 1, \cdots, 1, 1)\text{인 것의 개수를 } n(2^m - 1) \end{aligned} \tag{2.30}$$

라고 할 때, N개의 항으로 이루어진 유한 이진 수열 식 (2.31)이

$$s_0, s_1, \cdots, s_{t+2}, \cdots, s_{N-1} \tag{2.31}$$

완전한 랜덤 수열인 경우에는 $GF(2)^n$에 속해 있는 각 벡터가 식 (2.28)에 있는 n개의 벡터 중의 하나로 나타날 확률은 $\frac{n}{2^m}$이다. 따라서 통계량 식 (2.32)는

$$\begin{aligned} T &= \sum_{i=0}^{2^m-1} \frac{\left\{n(i) - \frac{n}{2^m}\right\}^2}{\frac{n}{2^m}} \\ &= \frac{2^m}{n} \sum_{i=0}^{2^m-1} n(i)^2 - n \end{aligned} \tag{2.32}$$

근사적으로 자유도가 $2^m - 1$인 χ^2분포를 따른다.
이제 $0 < \alpha < 1$ 인 α에 대하여

$$P(\chi^2 \geqq x_\alpha) = \alpha \tag{2.33}$$

인 실수 x_α를 χ^2분포표에서 구하면, 유의수준 α에 대하여 기각역은 $T > x_\alpha$이다. $\alpha = 0.01$에 대하여 식 (2.34)와 같을 때

$$P(\chi^2 > = x_\alpha) = 0.01 \tag{2.34}$$

T·의 값이 α보다 크면 주어진 수열은 랜덤(Random)하지 않다고 말할 수 있고 T·의 값이 α보다 작으면 이 테스트로는 랜덤(Rrandom)하지 않다고 판정할 수는 없다. 일반적으로, 모든 m에 대하여 일일이 이와 같은 테스트를 시행할 수는 없으므로 경우에 따라 m의 값을 적절히 택한다. 특히, $m = 1$인 경우에 이 테스트를 흔히 프리퀀시 테스트라고 한다. 이 경우에 식 (2.35)와 같다.

$$T = \frac{2}{N}\sum_{i=1}^{1}\left(n(i) - \frac{N}{2}\right)^2 \tag{2.35}$$

여기서 $n(0)$과 $n(1)$은 각각 $s_0, s_1, s_2, \cdots, s_{N-1}$중에서 0인 것의 개수와 1인 것의 개수를 뜻한다. 이제 $n_0 = n(0)$, $n_1 = n(1)$이라고 놓으면,

$$N = n = n(0) + n(1) = n_0 + n_1 \tag{2.36}$$

이므로, 식 (2.36)은 식 (2.37)로 변형된다.

$$T = \frac{(n_0 - n_1)^2}{N} \tag{2.37}$$

그리고, 이 경우에 통계량 식 (2.38)은

$$Z = \frac{1}{\frac{\sqrt{n}}{4}} \left\{ n_1 - \frac{n}{2} \right\} \tag{2.38}$$

근사적으로 표준정규분포 $N(0,1)$을 따른다. 한편, χ^2분포에서 유의수준 5%의 한계값은 3.84이다. 따라서 T·의 값이 3.84보다 큰 경우에, 프리퀀시 테스트에 대하여 유의수준 5%로 이 이진 수열은 난수성이 없다고 판단되어 기각한다. 이에 대한 판단으로 유의확률 $P-value$를 사용하는데 $P-value$ 0.01이상에 대하여 유의수준을 결정하며, [표 2.10]은 프리퀀시 테스트의 예 로 $P-value$ 값이 0.01 이상이므로 수열 e는 랜덤하다.

[표 2.10] 프리퀀시 테스트 예

input	$e =$ 1100100100001111110110101010001000100001011010001100001000110100110001001100011001100010100010111000
	$n = 100$
processing	$S_n = -16$
	$s_{obs} = 1.6$
output	$P-value = 0.109599$

3. 런(Run) 테스트

이진 수열 (s_t)에서 0또는 1이 처음부터 끝까지 반복해서 연이어 나타나는 부분을 이진 수열의 런 이라고 한다. 예를 들어, 0110001은 한 개의 0으로 이루어진 런과 두 개의 1로 이루어진 런, 세 개의 0으로 이루어진 런, 그리고 한 개의 1로 이루어진 런 을 포함하고 있다. 특히, 0만으로 이루어진 런을 그 수열의 갭(Gap) 이라 하고, 1만으로 이루어진 런을 그 수열의 블록(Block)이라 한다. 다음은 주기가 N인 이진 수열의 임의성에 관한 Golomb의 공리계이다.

$R1$ 먼저 N이 짝수인 경우에, 길이가 N인 순환마디에는 $\frac{N}{2}$개의 0과 $\frac{N}{2}$개의 1이 들어 있다. 한편, N이 홀수인 경우에는 0또는 1이 $[\frac{N}{2}]$개씩 들어 있다. $R2$길이가 N인 순환마디에 들어 있는 런 가운데 절반은 길이가 1이고 $\frac{1}{4}$은 길이가 2이며, 일반적으로 이 순환마디에 적어도 2^{i+1}개의 런이 들어 있다면 이 중에서 $\frac{1}{2^i}$은 길이가 i이다. 또한, 각 $i>1$에 대하여 길이가 i인 갭 과 블록의 개수는 동일하다. $R3$ 이 수열의 Out-of-Phase 자기상관은 일정하다. 위의 $R1, R2, R3$를 만족시키는 무한 이진 수열을 흔히 G-Random 수열 또는 PN 수열 이라고 한다. 이진 수열 (s_t)의 N개의 항 $(s_0, s_1, s_2, \cdots, s_{m-1})$에서 갭의 개수와 블록의 개수를 각각 r_0, r_1이라 하고 $n = r_0 + r_1$이라고 하자. 또, 각 $i(1 \leq i \leq L)$에 대하여 길이가 i인 갭의 개수와 길이가 i인 블록의 개수를 각각 n_{0i}, n_{1i}라 하고 $n_i = n_{0i} + n_{1i}$라고 하면, n은 런의 개수이고, n_i는 길이가 i인 런의 개수이다. 이때, 갭 중에서 그 길이가 i일 확률은 $\frac{1}{2^{i+2}}$이고 블록 중에서 그 길이가 i일 확률은 $\frac{1}{2^{i+2}}$이다. 따라서 통계량 식 (2.39)는

$$T = \sum_{j=0}^{1} \sum_{i=1}^{L} \frac{\left(n_i^k - \frac{n}{2^{i+2}}\right)^2}{\frac{n}{2^{i+2}}} \tag{2.39}$$

근사적으로 자유도가 $2L$인 카이제곱 분포를 따른다. 이와 같은 통계량을 이용하는 테스트를 매개변수가 L인 런 테스트라 하며, [표 2.11]은 런 테스트의 예 로 $P-value$ 값이 0.01 이상이므로 수열 e는 랜덤하다.

[표 2.11] 런 테스트 예

input	$e =$ 1100100100001111110110101010001000100001011010001100001 0001101001100010011000110011000101000101110 00
	$n = 100$
	$t = 0.02$
processing	$p = 0.42$
	$V_n(obs) = 52$
output	$P-value = 0.500798$

4. 시리얼(Serial) 테스트

시리얼 테스트는 한 항이 그 다음에 0또는 1로 바뀌어 나가는 과정이 랜덤 한지를 조사하는 방법이다[11].

이진 수열 (s_t)의 N개의 항 $s_0, s_1, s_2, \cdots, s_{N-1}$중에서 0인 것의 개수와 1인 것의 개수를 각각 n_0, n_1이라고 하자. 또 이들 N개의 항을 차례를 연이어 두 항씩 묶어 놓은

$$s_0s_1,\ s_1s_2,\ s_2s_3,\ \cdots,\ s_{N-2}s_{N-1}$$

중에서 00, 01, 10, 11과 같은 것의 개수를 각각 $n_{00}, n_{01}, n_{10}, n_{11}$이라고 하자. 이때, 다음 등식이 성립한다.

$$n_0 + n_1 = N$$

$$n_{00} + n_{01} = n_0 \text{ 또는 } n_0 - 1$$

$$n_{10} + n_{11} = n_1 \text{ 또는 } n_1 - 1$$

$$n_{00} + n_{01} + n_{10} + n_{11} = N - 1$$

한편, 각 n_{ij}의 기댓값은 $\frac{N-1}{4}$이다. 따라서 통계량 식 (2.40)은

$$T = \sum_{i,j=0}^{1} \frac{\left(n_{ij} - \frac{N-1}{4}\right)^2}{\frac{N-1}{4}} - \sum_{i=0}^{1} \frac{\left(n_i - \frac{N}{2}\right)^2}{\frac{N}{2}} \tag{2.40}$$

근사적으로 자유도가 2인 χ^2분포를 따른다.
식 (2.40)위 식은 다음과 같이 간단히 식 (2.41)과 같이 표현된다.

$$T = \frac{4}{n-1}(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2) \tag{2.41}$$

$$- \frac{2}{N}(n_0^2 + n_1^2) + 1$$

이 통계량을 이용하여 난수성을 테스트하는 방법을 시리얼 테스트라고 한다.
예를 들어 길이 10인 이진 수열 0011011101에 대하여 비트길이를 3까지로 제안 한다면 3비트씩 나누어야 하므로 끝에 00를 첨가한 001101110100을 이용하여 빈도를 측정한다. 같은 방법으로 2비트열에 대한 빈도 측정은 00110111010을 사용하고 단일비트에 대한 빈도수 측정은 프리퀀시 테스트와 동일한 방법으로 시행한다.

각 비트열에 대한 빈도수 측정에 대한 확률값은 n이 총도수, m =3인 비트 열수, v는 각 비트의 패턴 일때 식 (2.42), 식 (2.43), 식 (2.44)에 의해 산출된다.

$$\Psi_m^2 = \frac{2^m}{n} \sum_{i_1 \cdots i_m} \left(v_{i_1 \cdots i_m} - \frac{n}{2^m} \right)^2 \quad : \text{ (3비트 수식)} \tag{2.42}$$

$$= \frac{2^m}{n} \sum_{i_1 \cdots i_m} v_{i_1 \cdots i_m}^2 - n$$

$$\Psi_{m-1}^2 = \frac{2^{m-1}}{n} \sum_{i_1 \cdots i_{m-1}} \left(v_{i_1 \cdots i_{m-1}} - \frac{n}{2^{m-1}} \right)^2 \text{ : (2비트 수식)} \tag{2.43}$$

$$= \frac{2^{m-1}}{n} \sum_{i_1 \cdots i_{m-1}} v_{i_1 \cdots i_{m-1}}^2 - n$$

$$\Psi_{m-2}^2 = \frac{2^{m-2}}{n} \sum_{i_1 \cdots i_{m-2}} \left(v_{i_1 \cdots i_{m-2}} - \frac{n}{2^{m-2}} \right)^2 \text{ : (1비트 수식)} \tag{2.44}$$

$$= \frac{2^{m-2}}{n} \sum_{i_1 \cdots i_{m-2}} v_{i_1 \cdots i_{m-2}}^2 - n$$

식 (2.42), 식 (2.43), 식 (2.44)를 이용하면 식 (2.45), (2.46), (2.47)을 얻는다.

$$\Psi_3^2 = \frac{2^3}{10}(0+1+1+4+1+4+4+1) - 10 \tag{2.45}$$

$$= 12.8 - 10$$

$$= 2.8$$

$$\Psi_2^2 = \frac{2^2}{10}(1+9+9+9) - 10 \tag{2.46}$$

$$= 1.2$$

$$\Psi_1^2 = \frac{2}{10}(16+36) - 10 = 10.4 - 10 \tag{2.47}$$

$$= 0.4$$

여기서, χ^2분포에서 유의수준 5%, 자유도 2의 한계값은 5.99이다.

따라서 T의 통계량의 값이 5.99보다 큰 경우에, 시리얼 테스트에 대하여 유의수준 5%로 이 이진 수열은 난수성이 없다고 판단되어 기각하며, [표 2.12]는 시리얼 테스트의 예 로 $P-value1$, $P-value2$ 값이 0.01 이상이므로 ϵ는 랜덤하다.

[표 2.12] 시리얼 테스트 예

input	$\epsilon = 1{,}000{,}000$ 비트
	$m = 2;\quad n = 1{,}000{,}000 = 10^6$
processing	$\#0s = 499{,}971;\quad \#1s = 500{,}029$
	$\#00s = 250{,}116;\quad \#01s = \#10s = 249{,}855;\quad \#11s = 250{,}174$
	$\Psi_2^2 = 0.343128;\ \Psi_1^2 = 0.003364;\ \Psi_0^2 = 0.000000$
	$\nabla\Psi_2^2 = 0.339764;\quad \nabla^2\Psi_2^2 = 0.336400$
output	$P-value1 = 0.843764;\quad P-value2 = 0.561915$

5. 포커 테스트

이진 수열 (S_t)의 N개의 항 $s_0, s_1, s_2, \cdots, s_{N-1}$을 처음부터 차례로 m개씩 나누면, 다음과 같이 m차원 벡터공간 $GF(2)^2$에 속하는 $n = [\frac{N}{m}]$개의 식 (2.48)의 벡터가 생긴다.

$$\begin{aligned} &(s_0, s_1, \cdots, s_{t+2}, \cdots, s_{m-1}) \\ &(s_m, s_{m+1}, \cdots, s_{m+2}, \cdots, s_{2m-1}) \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ &(s_{(n-1)m}, s_{(n-1)m+1}, \cdots, s_{m+2}, \cdots, s_{nm}) \end{aligned} \tag{2.48}$$

이들 벡터 중에서 i개의 1과 $m-1$개의 0으로 이루어진 벡터의 개수를 $n(i)$라고 하자. 이 때, 수열 (s_t)가 랜덤하다면, 각 i $(1 \le i \le m)$에 대하여 $n(i)$의 평균값은 $\binom{m}{i}$ 이다.

따라서 통계량 식 (2.49)는 근사적으로 자유도가 m인 χ^2분포를 따른다.

$$T = \sum_{i=0}^{m} \frac{\left(n(i) - \binom{m}{i}\frac{n}{2^m}\right)^2}{\binom{m}{i}\frac{n}{2^m}} = \frac{2^m}{n}\sum_{i=0}^{m}\frac{n(i)^2}{\binom{m}{i}} - n \tag{2.49}$$

이와 같은 난수성 검정법을 포커 테스트(Poker test)라 한다. 이 검정법은 프리퀀시 엠 테스트에서 사용되는 통계량보다 단순하므로 이용하기에는 쉬우나 정밀성은 떨어진다.

Ⅲ. 양자암호

A. 양자암호

1. BB84 프로토콜

암호용 키 분배는 크게 두 가지로 나눌 수 있다. 비밀키를 담당자에게 배포하여 관리하는 것과 공개 키 분배방식이다. 전자는 사람에 대한 신뢰를 믿을 수가 없으며 후자는 소인수분해의 해법이 완성되면 안전성을 보장할 수 가 없다. 양자컴퓨터를 이용한 쇼의 소인수분해 알고리즘이 상용화되면 RSA공개키 암호기법은 근본적인 안정성에 문제가 발생한다. 양자역학의 불확정성을 이용한 키 분배방식은 도청자의 유무를 파악할 수 있기에 새로운 암호이론으로 각광받고 있다[4,30,31,32,35].

편광된 광자를 이용하는 양자암호 방식은 베넷(C. H. Bennett), 브라사드(G. Brassard)에 의해 1984년에 제안 되었다. BB84 프로토콜은 양자역학의 관측이론과 원타임패드(One-time-pad) 암호 방식을 결합하여 해독이 불가능하게 만든 암호 방식이다[2,13,33,34]. 가로와 세로로 직선편광된 $|\leftrightarrow\rangle$와 $|\updownarrow\rangle$상태, 대각방향 $+45^\circ$와 -45°로 편광된 $|\nearrow\rangle$와 $|\nwarrow\rangle$상태 등 총 네 종류의 광을 사용한다.

[표 3.1] 편광된 광자의 이진 대응표

비트값	$\oplus$	$\otimes$
0	$\|\updownarrow\rangle$	$\|\nwarrow\rangle$
1	$\|\leftrightarrow\rangle$	$\|\nearrow\rangle$

송신자 Alice와 수신자 Bob은 가로, 세로의 직선편광 광자와 대각선의 직선 편광 광자를 동시에 이용한다. 송신자는 $\oplus$와 $\otimes$두 종류의 편광필터를 무작위로 사용하여 비트를 송신하고 수신자도 두 종류의 검출기를 무작위로 사용하여 광을 검출한다. BB84 프로토콜은 다음과 같다.

- 송신자는 $\oplus$와 $\otimes$편광필터를 무작위로 선택하여 0과 1이 무작위로 배열된 $4n$ 비트 데이터를 송신한다.
- 수신자는 $\oplus$와 $\otimes$편광검출기를 무작위로 택하여 편광방향을 관측한다. 송신자는 수신자에게 자신이 선택한 편광필터의 배열 순서를 공개된 채널을 통해 알린다.
- 두 사람은 검출기의 $\oplus$와 $\otimes$종류와 송신자의 편광필터 $\oplus$와 $\otimes$가 일치하는 경우만 참값으로 인정하고 나머지는 버린다. 편광필터와 편광검출기가 일치할 확률은 $\frac{1}{2}$이므로 $2n$비트의 동일한 데이터를 공유하게 된다. 그중 n비트의 데이터를 상호 조합하여 확인하고 나머지 n비트를 이용하여 원타임패드 (One-time-pad) 를 만든다.
- 송신자는 평문을 n비트의 원타임패드(One-time-pad) 를 이용하여 암호화하고 이를 수신자에게 보낸다.
- 수신자는 받은 암호문을 공유하는 원타임패드(One-time-pad) 로 해독한다. 가로 세로 편광상태는 검출기의 대각편광으로 검출을 하면 $\frac{1}{2}$의 확률로 대각편광 상태로 관측된다. 만약 중간에 공격자가 가로채기를 하고 다시 수신자에게 신호를 보낸다면 이는 $\frac{1}{4}$이상의 오류를 보여주게 된다. 오류 상태가 정상적이지 않을 때는 첫 단계부터 다시 편광을 보내서 시작하면 된다.

[표 3.2]에서 나타난 바와 같이 송신자가 보내는 데이터에는 보내고자 하는 송신 비트들을 이진 비트가 아닌 편광 형태로 변형하여 무작위 선택한 편광기를 사용한다. 중간에 도청자가 새로운 검출기를 사용하여 편광을 복사하는 것은 이론상 불가능하므로 도청에 의한 편광복사는 존재할 수가 없다. 다만 송신자와 수신자가 사용하는 송신 비트와 편광기 선택 비트 그리고 수신자가 선택하는 검출기 선택비트들에서 실난수 사용상의 애로점으로 인하여 의사난수를 사용하므로 중간자공격(Man-in-the-Middle Attack)과 부분정보 유출에 대한 애로점은 존재한다고 볼 수 있다. BB84 프로토콜에 의하여 n개의 비트 값을 관찰하고 도청자를 발견할 확률은 각각의 비트들이 난수성을 확보했다는 가정 하에 다음과 같은 식 (3.1)의 계산결과를 볼 수 있다.

$$P(n) = 1 - \left(\frac{3}{4}\right)^n \tag{3.1}$$

이는 비트수가 많을수록 도청자의 유무를 판별하기가 수월해진다.

[표 3.2] BB84 프로토콜

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
송신자	송신비트	0	1	1	0	1	1	1	0	0	0	1	0	0	1	1	1	0	1	0	0
	필터	⊕	⊗	⊕	⊗	⊕	⊕	⊗	⊕	⊕	⊕	⊕	⊗	⊗	⊕	⊗	⊕	⊗	⊗	⊕	⊗
	상태	$\|\updownarrow\rangle$	$\|\nearrow\rangle$	$\|\leftrightarrow\rangle$	$\|\nwarrow\rangle$	$\|\leftrightarrow\rangle$	$\|\leftrightarrow\rangle$	$\|\nearrow\rangle$	$\|\updownarrow\rangle$	$\|\updownarrow\rangle$	$\|\updownarrow\rangle$	$\|\leftrightarrow\rangle$	$\|\nwarrow\rangle$	$\|\nwarrow\rangle$	$\|\leftrightarrow\rangle$	$\|\nearrow\rangle$	$\|\leftrightarrow\rangle$	$\|\nwarrow\rangle$	$\|\nearrow\rangle$	$\|\updownarrow\rangle$	$\|\nwarrow\rangle$

수신자	검출	⊕	⊕	⊗	⊗	⊕	⊕	⊗	⊕	⊗	⊗	⊗	⊕	⊕	⊗	⊕	⊗	⊗	⊗	⊕	⊕
	관측	$\|\updownarrow\rangle$	$\|\updownarrow\rangle$	$\|\nearrow\rangle$	$\|\nwarrow\rangle$	$\|\leftrightarrow\rangle$	$\|\leftrightarrow\rangle$	$\|\nearrow\rangle$	$\|\updownarrow\rangle$	$\|\nearrow\rangle$	$\|\nearrow\rangle$	$\|\nearrow\rangle$	$\|\updownarrow\rangle$	$\|\updownarrow\rangle$	$\|\nearrow\rangle$	$\|\updownarrow\rangle$	$\|\nearrow\rangle$	$\|\nwarrow\rangle$	$\|\nearrow\rangle$	$\|\updownarrow\rangle$	$\|\updownarrow\rangle$
	비트	0	0	1	0	1	1	1	0	1	1	1	0	0	1	0	1	0	1	0	0

일치	T	F	F	T	T	T	T	T	F	F	F	F	F	F	F	F	T	T	T	F
원타임 패드	0			0	1	1	1	0									0	1	0	

2. B92 프로토콜

BB84 프로토콜에서는 가로와 세로로 직선편광된 $|\leftrightarrow\rangle$와 $|\updownarrow\rangle$상태, 대각방향 $+45^\circ$와 -45°로 편광된 $|\nearrow\rangle$와 $|\nwarrow\rangle$상태 등 총 네 종류의 광을 사용하였다. 프로토콜의 특징은 두 개의 비 직교상태를 이용하여 비밀열쇠를 배분하는 방법이다. 이 프로토콜은 두 개의 비 직교상태를 이진법 0과 1에 대응시킨다. 편광된 광자를 이용한 경우에는 가로·세로향상태 $|\updownarrow\rangle$를 0에 대응시키고 -45° 방향의 편향상태$|\searrow\rangle$를 1에 대응시킨다[2].

[표 3.3] B92 프로토콜에 의한 편광상태와 비트 값을 사영한 측정기

송신자		수신자
비트값	편광상태	측정기
0	$\updownarrow$	$P_0 = 1 - \lvert\nwarrow\rangle\langle\nwarrow\rvert$
1	$\nwarrow$	$P_1 = 1 - \lvert\updownarrow\rangle\langle\updownarrow\rvert$

두 상태의 내적은 식 (3.2)로 표현된다.

$$\langle\nwarrow|\updownarrow\rangle = \frac{1}{\sqrt{2}} \tag{3.2}$$

수신자는 비트 값을 사영하는 P_0와 P_1의 두 가지 측정기를 가지는데 측정기 P_0는 $|\searrow\rangle$상태를 관측하지 못한 경우에 비트 값 0을 가지고, 측정기 P_1은 $|\updownarrow\rangle$상태를 측정하지 못한 경우 비트 값 1을 갖는다. 송신자가 비트 값 $|0\rangle = |\updownarrow\rangle$을 발신하면 식 (3.3)으로 표현되고

$$\begin{aligned} \langle\nwarrow|P_1|\updownarrow\rangle &= \langle\nwarrow|(|\updownarrow\rangle - |\updownarrow\rangle) \\ &= 0 \end{aligned} \tag{3.3}$$

이고 수신자의 측정결과 의 확률은 0이다. 한편

$$\begin{aligned} \langle\updownarrow|P_0|\updownarrow\rangle &= 1 - |\langle\nwarrow|\updownarrow\rangle|^2 \\ &= 1 - \frac{1}{2} \\ &= \frac{1}{2} \end{aligned} \tag{3.4}$$

으로 되어서 $\frac{1}{2}$의 확률도 비트 값 0을 측정한다. 비트 값 1에 대응하는 대각 편광상태 $|\searrow\rangle$에 대해서도 동일하게

$$< \updownarrow | P_0 | \nwarrow > \; = \; 0 \tag{3.5}$$

$$< \nwarrow | P_1 | \nwarrow > \; = \frac{1}{2} \tag{3.6}$$

로 되어서 비트 값 1이 $\frac{1}{2}$의 확률로 주어진다. B92 프로토콜의 특징은 두 종류의 광자를 사용하더라도 50%의 확률로 바른 값을 전달 가능하다. 수신자가 무작위로 측정기를 선택할 때 수신자의 신호가 전달될 확률은 식 (3.7)로 표현된다.

$$\frac{1}{2}(1 - < \updownarrow | \nwarrow >^2) = \frac{1}{4} \tag{3.7}$$

그러나 송신자의 비트 값이 잘못 전달될 확률은 0이라는 사실이 B92 프로토콜의 중요한 부분이다. 도청자가 송신자가 발신한 비트 값을 도청하고자 할 경우 편광 상태가 비 직교계 이므로 양자상태의 비 복제정리에 의해 도청자는 송신자의 상태를 복사할 수 없다. 즉 도청자는 송신자가 발신한 비트 값을 관측하여 그 관측 상태를 재발신하게 된다. 이것이 도청자를 발견하게 되는 요인이다. 일반적으로 BB84 프로토콜, B92 프로토콜 통신에는 평문을 보내는 비트와 도청자를 검출하는 비트의 두 종류를 사용할 필요가 있다.

3. E91 프로토콜

BB84 프로토콜, B92 프로토콜은 서로 직교하지 않는 양자상태는 복제불능이라는 이론에 의해 안전성이 보증된 비밀열쇠 분배법이다. 이들 두 프로토콜에는 양자역학의 불확정성 관계가 본질적 역할을 하고 있다. 두 입자가 상관된(Entangled) 양자상태 즉 EPR쌍을 이용하여 암호 열쇠를 안전하게 분배할 수 있다[2].

E91 프로토콜의 경우 송신자와 수신자는 스핀 0으로 결합한 두 입자계, EPR 쌍을 관측한다. EPR 쌍은 스핀 $\frac{1}{2}$인 입자를 이용하면 식 (3.8)을 얻을 수 있다.

$$|\Psi>_{12} = \frac{1}{\sqrt{2}}\{|\uparrow>_1|\downarrow>_2 - |\downarrow>_1|\uparrow>_2\} \tag{3.8}$$

따라서 EPR 쌍은 두 개의 광자로도 만들어질 수 있다. EPR 쌍은 z축 반대방향으로 발사되며 송신자와 수신자는 서로 떨어진 장소에서 각 각의 편광 검출기로 한 개의 입자의 스핀 방향을 관측한다. 단위 벡터 a_i와 b_j $(i=1,2,3)$ 로 검출기의 방향을 표시하며 이 두 단위 벡터가 z축에 수직한 xy평면에 위치하게 한다. 송신자가 a_i의 검출기를 이용하고 수신자는 b_j검출기를 이용하여 스핀의 방향을 측정하면 상행인 +방향과 하향인 −방향의 상관식은 식 (3.9)와 같이 a_i와 b_j의 내적으로 주어진다.

$$\begin{aligned} E(a_i,b_j) = & \; E_{++}(a_i,b_j) + E_{--}(a_i,b_j) - E_{+-}(a_i,b_j) - E_{-+}(a_i,b_j) \\ = & \; -(a_i,b_j) \end{aligned} \tag{3.9}$$

벨의 부등식의 한 종류인 CHSH 부등식(Clauser, Horne, Shimony, Holt 부등식)을 유도하기 위해 상관 함수

$$S = E(a_1,b_1) - E(a_1,b_3) + E(a_3,b_1) + E(a_3,b_3) \tag{3.10}$$

을 생각하자. 식 (3.10)을 구하려면 송신자와 수신자가 다른 방향에서 EPR 쌍을 지정된 방향으로 4회 측정해야 된다. 양자역학적 관측결과는

$$S_{QM} = -\cos(\phi_1 - \theta_1) + \cos(\phi_1 - \theta_3) - \cos(\phi_3 - \theta_1) - \cos(\phi_3 - \theta_3) \tag{3.11}$$

로 주어지며, E91 프로토콜은 다음과 같다.

단계 1 : 송신자와 수신자는 EPR 쌍을 측정기를 이용하여 관측한다.
단계 2 : 송신자와 수신자는 검출기 방향을 공개하고 측정결과를
(a) 다른 방향의 검출기를 이용한 결과
(b) 같은 방향의 검출기를 이용한 결과

의 두 종류로 분류한다.

단계3 : 송신자와 수신자는 (a) 그룹에 속하는 결과만 공개하고 CHSH 부등식 S가 만족되는지를 확인한다.

단계4 : CHSH 부등식 $S = -2\sqrt{2}$가 만족되는 경우에만 (b) 그룹의 결과를 원타임 패드(One-time pad)용 비밀열쇠로 사용한다.

EPR 쌍의 관측결과로부터 송신자와 수신자는 E91 프로토콜에 의해 비밀 열쇠를 분배했다고 하자. 도청자가 E91 프로토콜을 부수고 비밀 열쇠를 훔치고자 할 때 CHSH 부등식은 어떻게 변화하는가가 문제이다. 도청자가 EPR 쌍으로부터 정보를 훔치려고 할 경우의 CHSH 부등식을 유도하면 E91 프로토콜의 제 1특징은 도청자가 가령 송신자와 수신자가 관측하고자 하는 모든 EPR 쌍을 관측하였다 하더라도 어떤 정보도 얻을 수 없다는 것이다. 즉 비밀 열쇠인 EPR 쌍 자체가 아니라 송신자와 수신자가 이 관측을 한 후 단계 2의 (b)데이터만을 선택하여 만들어지기 때문이다. 도청자가 암호열쇠의 정보를 얻으려면 EPR 쌍을 관측한 후 다시 EPR 쌍을 송신자와 수신자에게 발신하여 두 사람의 검출기 종류를 알아야 한다. 그래서 도청자는 두 대의 검출기 A, B를 이용하여 ϕ_A와 θ_B방향에서 EPR 쌍의 스핀을 관측하고 그 관측한 스핀과 같은 방향으로 다시 두 개의 입자를 발신한다고 생각해 보자. 도청자가 ϕ_A와 θ_B방향을 임의 확률 $P(\phi_A, \theta_B) = |a(\phi_A,\ \theta_B)|^2$로 택했다고 할 때 발신된 두 입자의 상태는 진폭 $a(\phi_A,\ \theta_B)$를 이용하여 식 (3.12)로 표현된다.

$$|\Psi(A,\ B) \geqq \int_0^{2\pi} d\phi_A \int_0^{2\pi} d\theta_B\ a(\phi_A, \theta_B)|\phi_A >_1 |\theta_B >_2 \tag{3.12}$$

여기서 직교규격화 되어있으므로 $|\Psi(A,\ B) >$도 식 (3.13)으로 규격화

$$< \Psi(A,\ B)|\Psi(A,\ B) > \ = \int_0^{2\pi} \int_0^{2\pi} |a(\phi_A, \theta_b)|^2 d\phi_A d\theta_B = 1 \tag{3.13}$$

된다. 도청자가 발신한 $|\phi_A >, |\theta_B >$상태를 송신자와 수신자는 검출기 a와 b를 이용하여 그 스핀방향을 관측한다면 도청자가 존재하는 경우 식 (3.14)로 표현된다.

$$E^{'}(a,b) = \iint p(\phi_A, \theta_B)\cos(\phi - \phi_A)\cos(\theta - \theta_B)d\theta_A d\theta_B \tag{3.14}$$

도청자가 도청한 결과 상관함수는 식 (3.15)로 되며,

$$-\sqrt{2} \le S^{'} \le \sqrt{2} \tag{3.15}$$

양자역학의 원리로부터 유도된 결과인 $S_{QM} = -2\sqrt{2}$ 와 모순이 된다. 이렇게 일반화된 벨의 정리, CHSM 부등식을 이용하여 송신자와 수신자는 도청자의 존재 여부를 알 수 있게 된다.

도청자가 EPR 쌍을 관측하는 것은 스핀의 방향을 확인하려는 것이다. 즉 관측에 따라 양자역학적인 2체계 상태가 수축하여 어떤 특정 스핀 방향이 정해지게 되기 때문이다. 즉 도청자가 존재하면 EPR 쌍이 교란을 받아 고전적 벨의 부등식이 성립하게 된다는 의미이다. 몇 가지 다른 방향에 있는 검출기 측정 결과로부터 도청자의 존재를 알 수 있는 이유는 서로 얽힌 상태를 설명해주는 양자역학의 원리를 잘 이용하기 때문이다.

Ⅳ. 양자암호 취약성 및 개선

본 장에서는 BB84 프로토콜 상에서 난수 병합에 따른 양자암호 취약성을 제시하고 문제점을 개선한 양자암호 사용에서 의사난수를 생성하지 않는 방안 즉 실난수를 사용할 수 있는 광자 발생 시스템의 의사코드 알고리즘을 제안하며 양자암호시스템 내에서 사용되는 벡터들과 이로 인한 실난수의 생성원리를 양자이론을 적용하여 양자 비트 발생의 난수성을 입증하였다.

A. 이진병합에 의한 양자암호 취약성

고전암호 체계에서 발전하여 기계식 암호가 나왔으며 세계대전을 거치면서 보다 깊고 인간의 계산 영역을 넘어서는 기계를 개발하였다. 이후 전산학의 연산 영역을 넘어서는 계산이론에 의한 공개키 암호 방식이 도출되었다. 한동안 계산량의 안전성에 머물러 있던 공개키 암호 방식이 쇼어의 양자이론을 이용한 소인수분해 해법을 찾음으로 인해서 안전성에 문제가 발생하게 되었다.

이에 광자를 이용한 양자암호 방식의 제안으로 중간 도청자로부터 안전한 암호체계인 BB84 프로토콜이 제안되었다. 모든 시스템은 사용자의 편리성을 찾아 가다보면 신뢰도에 어느 정도 약점을 보이기 마련이다. 양자암호 시스템도 실난수의 사용에 불편함을 이용하여 송신자와 수신자의 의사난수 사용은 난수성의 도출에 취약성을 발견하는 경우를 볼 수 있다. 기본적으로 의사난수로써의 가치를 인정받기 위하여 NIST의 난수성 테스트를 이진 비트열 상태에서 통과하여야 한다. 하지만 난수성 테스트를 통과한 비트들이라도 다중 비트열들의 병합과정에서 병합 비트열들의 난수성을 온전히 보전되지 않는 경우를 발견할 수 있다.

[표 3.2] BB84 프로토콜에서 송신자의 입력신호를 하나의 비트열로 간주하고 편광필터를 고르는 무작위성을 다른 하나의 이진 수열로 볼 수 있다.

$a_1 =$ 10101101111001010010 송신 비트열

$a_2 =$ 01101110001001110100 편광필터 선택 비트열에 대하여

$$a_1 * a_2 = \binom{1}{0}\binom{0}{1}\binom{1}{1}\binom{0}{0}\binom{1}{1}\binom{1}{1}\binom{0}{1}\binom{1}{0}\binom{1}{0}\binom{1}{0}$$

$$\binom{1}{1}\binom{0}{0}\binom{0}{0}\binom{1}{1}\binom{0}{1}\binom{1}{1}\binom{0}{0}\binom{0}{1}\binom{1}{0}\binom{0}{0}$$

과 같이 나타낼 수 있다. 이는 의사난수 기본 테스트인 런 테스트, 모노비트 테스트, 시리얼 테스트를 모두 통과하는 결과를 볼 수 있다. 상기 이진 수열들의 송신자와 수신자의 상호 선택에 의한 무작위 이진 수열들을 살펴보면 최소한 3개 이상의 의사난수 비트열을 유추할 수 있다. 이에 3개의 이진 비트열을 상호 일치하는 데이터 쌍들의 진행과 불일치하는 데이터의 쌍으로 이진화하여 비교해 보면 이들은 난수성을 잃어버리는 경우를 바로 볼 수 있다.

$b_1 =$ 11001101000110100011 수신자 검출기 선택 비트열

$$a_1 * a_2 * b_1 = \binom{1}{0}\binom{0}{1}\binom{1}{1}\binom{0}{0}\binom{1}{1}\binom{1}{1}\binom{0}{1}\binom{1}{0}\binom{1}{0}\binom{1}{1}$$

$$\binom{1}{0}\binom{0}{0}\binom{0}{0}\binom{1}{1}\binom{0}{1}\binom{1}{1}\binom{0}{0}\binom{0}{1}\binom{1}{0}\binom{0}{0}$$

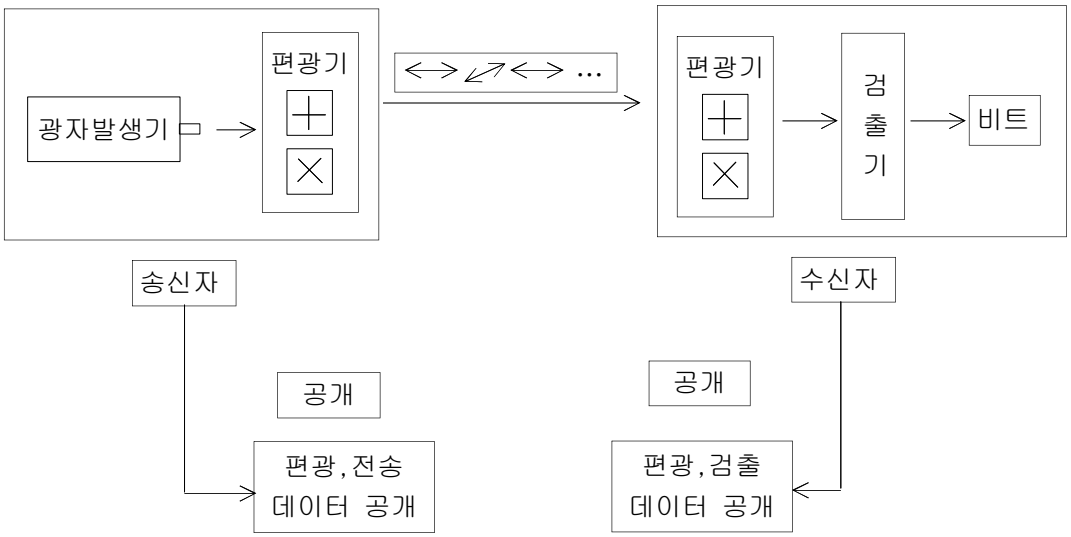
즉 송신 비트열과 수신자의 검출기 일치 여부를 연결하는 이진 비트열을 완성하면 다음과 같이 난수성을 잃어버린다.

$c =$ 10011111000000001110

본 논문에서는 의사난수열의 기본 테스트 과정과 이를 벗어나는 이진병합 비트열

의 난수성에 대하여 제안하였다. 송신자에 의해 생성된 이진 비트열 중 송신 비트와 필터 생성 비트열을 수신자에게 공개한 후 수신자의 검출 필터 생성 비트열과의 3중 비트 결합을 시키면 이진 수열 c=10011111000000001110 과 같은 일치 비트열을 얻을 수 있다. 이는 1과 0의 균등분포를 검출하는 프리퀀시 테스트는 통과함을 알 수 있으나 세번째 런과 네번째 런의 결과는 1과 0의 규칙적 패턴으로 확연히 런 테스트의 실패임을 알 수 있다. 암호시스템 내에서 작은 데이터 양이라도 부분 정보의 유출은 심각한 피해를 초래 할 수 있다. 실질적으로 구현하는 BB84 양자암호 시스템 내에서 난수성이 파괴된다는 것은 데이터 이동 경로상의 안전성이 입증이 되더라도 오픈채널에서 전위 비트열을 공유하는 한 의사난수의 안전성을 확보하지 못한다면 부분 정보 노출에 대하여 취약할 수밖에 없다. 보다 안전한 양자암호 시스템들이 개발되고 있는 현재 시점에서 의사난수열을 사용하는 시스템의 트렙도어를 제거해야하는 노력이 필요하고 이진 비트열들의 이진 병합에 대한 의사난수성을 입증하는 테스트까지 통과하는 의사난수 비트열들에 대해서만 난수열의 사용을 허용하거나 실난수를 사용할 수 있는 인프라 구축이 필요하다고 본다[9].

본 연구에서는 광자 발생기에 의한 양자 회절판을 이용해 [그림 4.1]의 BB84 프로토콜을 개선하여 사용되는 시스템 내에서 실난수를 생성하고 이를 이용한 양자암호 전송 프로토콜을 제안한다.

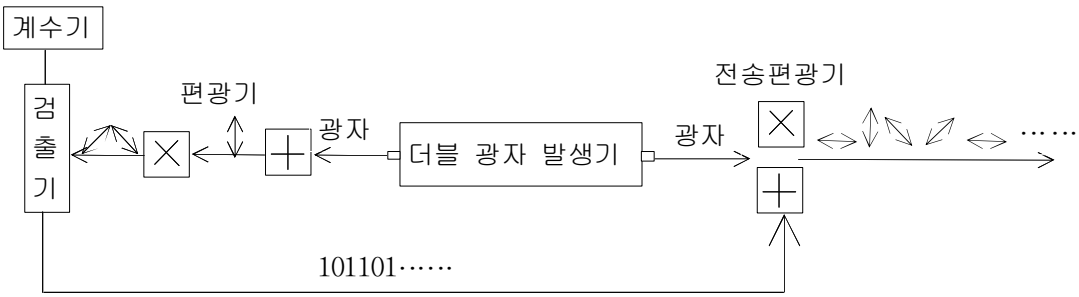


[그림 4.1] BB84 구조도

BB84 프로토콜에 대한 구조를 살펴보면 송신부의 광자 발생기에 의하여 생성된 광자는 랜덤 선택된 편광기를 거치게 된다. 이러한 랜덤 선택은 주로 의사난수를 이용하여 진행하는 경향이 강하다. +편광기와 ×편광기에서 생성된 벡터형태의 광자들이 전송되고 이러한 벡터들은 다시 수신부의 랜덤 선택된 편광기에 의하여 랜덤하게 검출기를 통과한다. 이후 공개된 채널을 통하여 송신자의 선택된 편광기와 편광기를 통과한 벡터들의 상태 즉 이진 비트로 표현 가능한 데이터들의 비트열을 동시에 공개한다. 수신자는 마찬가지로 임의 선택된 편광기와 이로 인한 벡터들의 비트 표현을 공개된 채널을 통하여 보여진다. 이는 모든 공격가능자들로 하여금 송신자와 수신자의 의사난수열과 이로 인한 데이터의 일치관계를 부분적으로나마 알 수 있는 요인이 될 수 있다.

B. 더블 광자 발생기를 이용한 실난수 생성

송신자와 수신자의 의사난수열과 이로 인한 데이터의 일치관계를 부분적으로 알 수 있는 상태의 최소화를 위하여 송신부와 수신부의 편광기 선택을 의사난수가 아닌 실난수를 사용할 수 있게 광자발생단계에서 이중으로 광자를 발생하여 한 방향은 벡터 데이터를 전송하고 이러한 벡터 데이터 전송에 사용되는 편광기 선택에 있어 사용되는 난수열을 +편광기와 ×편광기의 벡터 일차결합에 의한 $\frac{1}{2}$의 확률로의 도출을 이용하여 실난수를 적용한 편광기 선택을 사용하는 것을 보여준다.



[그림 4.2] 실난수 생성기 송신부 구조도 1

- 더블 광자 발생기에서 우측 초기 128회의 광자다발은 버린다.
- 좌측 초기 128회의 광자다발을 이용하여 ⊞ 편광기에 투영시킨다.
- 생성되는 벡터들을 ⊠ 편광기로 재 투영 시킨다. 이로 인하여 45° 편광상태인 ⤡과 ⤢이 일차결합의 원리를 이용하여 랜덤하게 도출된다.
- 검출기에서 ⤡과 ⤢로 나타나는 벡터를 이진 비트로 변환시켜 스택에 전송한다.
- 계수기의 카운터가 128을 나타내면 생성된 이진 난수열을 전송편광 선택기로 보낸다.
- 전송편광 선택기의 스택에서 넘어온 128비트를 이용하여 ⊞, ⊠ 편광기를 랜덤 선택한다.
- 발생된 광자다발을 선택된 편광기로 벡터 전송한다.

[그림 4.2]는 BB84 프로토콜의 개선으로 더블 광자 발생기를 사용한 초기 난수열을 생성하는 과정을 보여준다. 더블 광자 발생기에서 n개의 광자를 발생시킨 후 + 검출판에 생성된 광자를 ×검출판에 재 투영 하고, 투영된 광자의 비트 변이를 128비트 씩 스택에 저장된 이진비트를 이용 128비트 씩 스택에 저장된 이진비트를 이용하여 검출판을 선택한 후 양자암호를 전송한다. 더블 광자 발생기를 적용한 의사코드 알고리즘은 [그림 4.3]과 같다.

// $\boxplus$: + 편광기, $\boxtimes$: × 편광기 //

R : $|r_1>|r_2>|r_3>|r_4>\ldots|r_{128}>$ 우광자 발생 // 첫 번째 버림 //

L : $|l_1>|l_2>|l_3>|l_4>\ldots|l_{128}>$ 좌광자 발생

in L to $\boxplus$

out Q : $|\leftrightarrow>|\leftrightarrow>|\leftrightarrow>|\leftrightarrow>\ldots|\leftrightarrow>$

q_1 q_2 q_3 q_4 q_{128}

in Q to $\boxtimes$

out G : $|\swarrow\!\!\nearrow>$ $|\nwarrow\!\!\searrow>$ $|\swarrow\!\!\nearrow>$ $|\swarrow\!\!\nearrow>\ldots$ $|\swarrow\!\!\nearrow>$

g_1 g_2 g_3 g_4 g_{128}

in G to bit generator.

for $i=1, 128$

$b_i = T_r(g_i)$ // $T_r(\ )$: 비트 변환 //

end

R : $|r_1>|r_2>|r_3>|r_4>\ldots|r_{128}>$

for $i=1, 128$

if $b_i=1$ *get* $\boxplus$ *in* r_i *to* $\boxplus$

else *get* $\boxtimes$ *in* r_i *to* $\boxtimes$

end

[그림 4.3] 더블 광자 발생기를 적용한 의사코드 알고리즘

상기 알고리즘을 적용한 실난수 생성 데이터 흐름은 [표 4.1]과 같다.

[표 4.1] 더블 광자 실난수 생성부(32비트 적용)

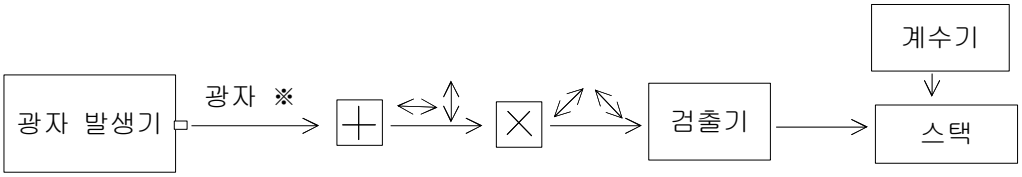
counter	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
우1광자	※	※	※	※	※	※	※	※	※	※	※	※	※	※	※	※
좌1광자	※	※	※	※	※	※	※	※	※	※	※	※	※	※	※	※
	＋ 편광기															
＋ 광자	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔
	× 편광기															
난수 생성	⤢	⤡	⤢	⤡	⤢	⤢	⤢	⤡	⤡	⤢	⤢	⤡	⤡	⤡	⤡	⤢
비트 변환	1	0	1	0	1	1	1	0	0	1	1	0	0	0	0	1
	전송편광기 선택															
	⊞	⊠	⊞	⊠	⊞	⊞	⊞	⊠	⊠	⊞	⊞	⊠	⊠	⊠	⊠	⊞
우1전송 광자	↔	⤢	↔	⤡	↕	↔	↕	⤡	⤢	↕	↕	⤢	⤡	⤢	⤢	↕
좌2	비트 생성부 순환															

counter	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
우1광자	※	※	※	※	※	※	※	※	※	※	※	※	※	※	※	※	버림
좌1광자	※	※	※	※	※	※	※	※	※	※	※	※	※	※	※	※	비트생성부
	+ 편광기																
+ 광자	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	
	× 편광기																
난수 생성	⤡	⤢	⤡	⤢	⤡	⤡	⤢	⤡	⤢	⤢	⤡	⤡	⤢	⤡	⤢	⤢	
비트 변환	0	1	0	1	0	0	1	0	1	1	0	0	1	0	1	1	
	전송편광기 선택																
	⊠	⊞	⊠	⊞	⊠	⊠	⊞	⊠	⊞	⊞	⊠	⊠	⊞	⊠	⊞	⊞	
우1전송 광자	⤡	↕	⤢	↕	⤢	⤢	↔	⤡	↕	↔	⤡	⤢	↔	⤢	↕	↔	
좌2	비트 생성부 순환																

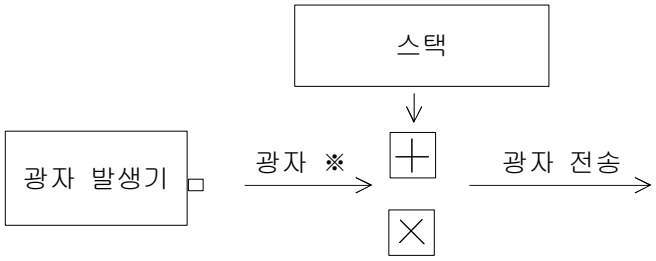
C. 싱글 광자 발생기를 이용한 실난수 생성

송신부와 수신부의 편광기 선택을 싱글 광자 발생기를 이용해 광자를 발생하여 광자열의 선택에 있어 사용되는 난수열을 +편광기와 ×편광기의 벡터 일차결합에 의한 $\frac{1}{2}$의 확률로의 도출을 이용하여 실난수를 적용한다. 이를 이용하여 편광기를 선택하고 이후 다시 전송광자를 발생한 후 기 생성된 편광기에 의하여 광자 전송을 마무리 하는 방법이다.

더블 광자 발생기에 비하여 하드웨어적 효율성이 뛰어남을 볼 수 있으나 데이터 흐름의 갭이 발생한다.



- 1 라운드 광자 발생 -



- 2 라운드 광자 전송 -

[그림 4.4] 실난수 생성기 송신부 구조도 2

- 싱글 광자 발생기에서 1라운드 128회의 광자다발을 이용하여 ⊞ 편광기에 투영 시킨다.
- 생성되는 벡터들을 ⊠ 편광기로 재투영 시킨다. 이로 인하여 45° 편광상태인 ⤡과 ⤢이 일차결합의 원리를 이용하여 랜덤하게 도출된다.
- 2라운드 128회의 광자다발을 이용하여 1라운드에서 생성된 난수열에 의하여 편광기를 선택한다.
- 검출기에서 ⤡과 ⤢로 나타나는 벡터를 이진 비트로 변환시켜 스택에 전송한다.
- 계수기의 카운터가 128을 나타내면 생성된 이진 난수열을 전송편광 선택기로 보낸다.
- 전송편광 선택기의 스택에서 넘어온 128비트를 이용하여 ⊞, ⊠ 편광기를 랜덤 선택 한다.

- 발생된 광자다발을 선택된 편광기로 벡터 전송한다.

[그림 4.4]는 BB84 프로토콜의 개선으로 싱글 광자 발생기를 사용한 초기 난수열을 생성하는 과정을 보여준다. 싱글 광자 발생기에서 128개의 광자를 발생시킨 후 +검출판에 생성된 광자를 ×검출판에 재 투영 하고, 투영된 광자의 비트 변이를 128비트 씩 스택에 저장된 이진비트를 이용하여 검출판을 선택한 전송광자를 발생시켜 양자암호를 전송한다. 싱글 광자 발생기를 적용한 의사코드 알고리즘은 [그림 4.5]와 같다.

P : $|p_1> |p_2> |p_3> |p_4> \ldots |p_{128}>$

$in\ P\ to\ \boxplus$

$out\ Q$: $|\leftrightarrow> |\leftrightarrow> |\leftrightarrow> |\leftrightarrow> \ldots |\leftrightarrow>$

$q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_{128}$

$in\ Q\ to\ \boxtimes$

$out\ G$: $|\swarrow\!\!\nearrow> |\swarrow\!\!\nearrow> |\nwarrow\!\!\searrow> |\swarrow\!\!\nearrow> \ldots |\nwarrow\!\!\searrow>$

$g_1 \quad g_2 \quad g_3 \quad g_4 \quad g_{128}$

$in\ G\ to\ bit\ generator.$

$for\ i = 1, 128$

$b_i = T_r(g_i)$ // $T_r(\ )$: 비트 변환 //

$if\ b_i = 1 \quad get\ \boxplus$

$else \quad get\ \boxtimes$

end

[그림 4.5] 싱글 광자 발생기를 적용한 의사코드 알고리즘

상기 알고리즘을 적용한 실난수 생성 데이터 흐름은 [표 4.2]와 같다.

[표 4.2] 싱글 광자 실난수 생성부(32비트 적용)

counter	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
광자	※	※	※	※	※	※	※	※	※	※	※	※	※	※	※	※
	+ 편광기															
+광자	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔
	× 편광기															
난수 생성	⤢	⤡	⤢	⤡	⤢	⤢	⤢	⤡	⤡	⤢	⤢	⤡	⤡	⤡	⤡	⤢
비트 변환	1	0	1	0	1	1	1	0	0	1	1	0	0	0	0	1
	전송편광기 선택															
	⊞	⊠	⊞	⊠	⊞	⊞	⊞	⊠	⊠	⊞	⊞	⊠	⊠	⊠	⊠	⊞
2라운드 광자전송																
광자 전송	↔	⤢	↔	⤡	↕	↔	↕	⤡	⤢	↕	↕	⤢	⤡	⤢	⤢	↕

counter	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
광자	※	※	※	※	※	※	※	※	※	※	※	※	※	※	※	※	
	+ 편광기																비트생성부
+광자	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	↔	
	× 편광기																
난수 생성	⤡	⤢	⤡	⤢	⤡	⤡	⤢	⤡	⤢	⤢	⤡	⤡	⤢	⤡	⤢	⤢	
비트 변환	0	1	0	1	0	0	1	0	1	1	0	0	1	0	1	1	
	전송편광기 선택																
	⊠	⊞	⊠	⊞	⊠	⊠	⊞	⊠	⊞	⊞	⊠	⊠	⊞	⊠	⊞	⊞	
2라운드 광자전송																	
광자 전송	⤡	↕	⤢	↕	⤢	⤢	↔	⤡	↕	↔	⤡	⤢	↔	⤢	↕	↔	

D. 양자 비트 발생의 난수성 입증

양자암호 시스템 내에서 사용되는 벡터들과 이로 인한 실난수의 생성원리를 양자이론을 적용하면 다음과 같다.

$|\uparrow>$는 캐트벡터라 하고 $<\uparrow|$는 브라벡터라 한다. 캐트벡터와 브라벡터는 n차원 복소벡터공간의 원소이고 서로 잉여관계에 있다. 먼저 각각 벡터들의 연결되는 사항을 일차결합을 이용하여 선형관계를 설명하고 스핀업과 스핀다운에 의해 설계된 벡터들을 크로스 벡터로 변환하는데 있어 고유벡터를 산출한 일차결합으로 표현하였다. 이는 스핀업과 스핀 다운의 벡터들이 크로스 벡터로 변환하는 과정이 $\frac{1}{2}$의 확률로 도출됨을 보여준다.

$$|\uparrow\rangle\langle\uparrow| = \begin{pmatrix}1\\0\end{pmatrix}(1\ 0) = \begin{pmatrix}1 & 0\\0 & 0\end{pmatrix}, \qquad |\uparrow\rangle\langle\downarrow| = \begin{pmatrix}1\\0\end{pmatrix}(0\ 1) = \begin{pmatrix}0 & 1\\0 & 0\end{pmatrix}$$

$$|\downarrow\rangle\langle\uparrow| = \begin{pmatrix}0\\1\end{pmatrix}(1\ 0) = \begin{pmatrix}0 & 0\\1 & 0\end{pmatrix}, \qquad |\downarrow\rangle\langle\downarrow| = \begin{pmatrix}0\\1\end{pmatrix}(0\ 1) = \begin{pmatrix}0 & 0\\0 & 1\end{pmatrix} \tag{4.1}$$

식 (4.1)과 같이 2행 2열로 표현되므로 스핀연산자는 식 (4.2)와 캐트벡터와 브라벡터의 텐서곱으로 고쳐 쓸 수 있다.

$$\sigma_x = |\uparrow\rangle\langle\downarrow| + |\downarrow\rangle\langle\uparrow|$$

$$\sigma_y = -i(|\uparrow\rangle\langle\downarrow| - |\downarrow\rangle\langle\uparrow|) \tag{4.2}$$

$$\sigma_z = (|\uparrow\rangle\langle\uparrow| - |\downarrow\rangle\langle\downarrow|)$$

스핀연산자 S의 크기는 식 (4.3)으로 표현되며

$$S_x = \begin{pmatrix}0 & \frac{\hbar}{2}\\ \frac{\hbar}{2} & 0\end{pmatrix} = \frac{\hbar}{2}\sigma_x$$

$$S_y = \begin{pmatrix}0 & -i\frac{\hbar}{2}\\ i\frac{\hbar}{2} & 0\end{pmatrix} = \frac{\hbar}{2}\sigma_y \tag{4.3}$$

$$S_z = \begin{pmatrix}\frac{\hbar}{2} & 0\\ 0 & -\frac{\hbar}{2}\end{pmatrix} = \frac{\hbar}{2}\sigma_z$$

여기서 σ_x, σ_y, σ_z는 식 (4.4)로 정의되고

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},\ \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{4.4}$$

다음과 같은 성질을 만족한다.

$$\sigma_x\sigma_y = i\sigma_z = -\sigma_y\sigma_x$$

$$\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{4.5}$$

$$\sigma_y\sigma_z = i\sigma_x = -\sigma_z\sigma_y$$

즉 $\sigma_i\sigma_j = \sigma_{ij}E + i\epsilon_{ijk}\sigma_k$.이다.
$(\sigma_1, \sigma_2, \sigma_3)$를 $\vec{\sigma}$로 정의하면 스핀행렬 $\vec{S}$는

$$\vec{S} = \frac{\hbar}{2}\sigma$$

$$S^2 = S_x^2 + S_y^2 + S_z^2 = \frac{3}{4}\hbar^2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$이 된다.

$$\begin{aligned} S^2 &= s(s+1) = \frac{1}{4}(\sigma_x^2 + \sigma_y^2 + \sigma_z^2) \\ &= \frac{3}{4} = \frac{1}{2}(\frac{1}{2}+1) \end{aligned}$$

으로부터 분명하게 $\frac{1}{2}$임을 알 수 있다. 또 z축 방향을 향하는 벡터는 식 (4.6)으로 표현된다.

$$s_z|\uparrow> = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{4.6}$$

$$s_z|\downarrow> = \frac{1}{2}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}0\\ 1\end{pmatrix} = -\frac{1}{2}\begin{pmatrix}0\\ 1\end{pmatrix}$$

여기서 $|\uparrow>$는 z축 양의 방향, $|\downarrow>$는 z축 음의 방향을 향하고 있음을 나타낸다.

3차원 벡터의 스핀 $\frac{1}{2}$인 상태를 단위벡터 $\vec{n}$을 식 (4.7)로 놓으면

$$\vec{n} = (\sin\theta\cos\varphi,\ \sin\theta\sin\varphi,\ \cos\theta) \tag{4.7}$$

$\vec{n}$방향 성분인 $\vec{S}\cdot\vec{n}$의 고유벡터는 식 (4.8)이므로

$$\begin{aligned}\vec{S}\cdot\vec{n} &= \frac{\hbar}{2}(\vec{\sigma}\cdot\vec{n})\\ &= \frac{\hbar}{2}\left\{\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}sin\theta\cos\varphi + \begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}sin\theta\sin\varphi + \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}cos\theta\right\}\\ &= \frac{\hbar}{2}\begin{pmatrix}\cos\theta & e^{-i\varphi}\sin\theta\\ e^{i\varphi}\sin\theta & -\cos\theta\end{pmatrix}\end{aligned} \tag{4.8}$$

고유값 식 (4.9)를 얻는다.

$$|\vec{S}\cdot\vec{n} - \lambda E| = 0$$

$$\begin{vmatrix}\frac{\hbar}{2}cos\theta - \lambda & \frac{\hbar}{2}e^{-i\varphi}\sin\theta\\ \frac{\hbar}{2}e^{i\varphi}\sin\theta & -\frac{\hbar}{2}cos\theta - \lambda\end{vmatrix} = 0 \tag{4.9}$$

$$\lambda^2 - \frac{\hbar^2}{4}cos^2\theta - \frac{\hbar^2}{4}sin^2\theta = 0$$

$$\therefore \lambda = \frac{\hbar}{2}, -\frac{\hbar}{2}$$

$\lambda = \frac{\hbar}{2}$인 경우 고유벡터 $u_+(\vec{n}) = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$는 식 (4.10)을 얻는다.

$$\begin{pmatrix} \frac{\hbar}{2}cos\theta - \frac{\hbar}{2} & \frac{\hbar}{2}e^{-i\varphi}\sin\theta \\ \frac{\hbar}{2}e^{i\varphi}\sin\theta & -\frac{\hbar}{2}cos\theta - \frac{\hbar}{2} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = 0$$

$$\frac{c_1}{c_2} = \frac{e^{-i\varphi}\sin\theta}{1-\cos\theta} = e^{-i\varphi}\frac{\cos\frac{\theta}{2}}{\sin\frac{\theta}{2}} \tag{4.10}$$

$$\text{즉 } u_+(\vec{n}) = \begin{pmatrix} e^{-i\varphi/2}\cos\frac{\theta}{2} \\ e^{i\varphi/2}\sin\frac{\theta}{2} \end{pmatrix}$$

마찬가지로 $\lambda = -\frac{\hbar}{2}$인 경우 식 (4.11)을 얻는다.

$$u_-(\vec{n}) = \begin{pmatrix} -e^{-i\varphi/2}\sin\frac{\theta}{2} \\ e^{i\varphi/2}\cos\frac{\theta}{2} \end{pmatrix} \tag{4.11}$$

일반적으로 스핀벡터는 식 (4.12)와 같이 표현되며

$$|\psi> = a|\uparrow> + b|\downarrow> = \begin{pmatrix} a \\ b \end{pmatrix} \tag{4.12}$$

이 상태는 스핀이 업일 확률 $|a|^2$과 다운일 확률 $|b|^2$로 나타난다.

$$|\uparrow_x> = a|\uparrow_z> + b|\uparrow_z> = \begin{pmatrix} a \\ b \end{pmatrix} \tag{4.13}$$

식 (4.13)을 규격화조건으로부터 식 (4.14)로 표현되며

$$<_x \uparrow | \uparrow_x> = |a|^2 + |b|^2 = 1 \tag{4.14}$$

식 (4.15)로 부터 식 (4.16)을 얻는다.

$$\sigma_x | \uparrow_x > = | \uparrow_x > \tag{4.15}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} b \\ a \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} \tag{4.16}$$

즉 식 (4.17)이 구해진다.

$$a = b \tag{4.17}$$

a를 양의 실수라 하면 x축 방향을 향한 상태는 식 (4.18)로 표현된다.

$$| \uparrow_x > = \frac{1}{\sqrt{2}} \{ | \uparrow_z > + | \downarrow_z > \} \tag{4.18}$$

또 $| \uparrow_x >$ 인 경우에는 식 (4.19)이므로 식 (4.20)과 같다.

$$\sigma_x | \downarrow_x > = - | \downarrow_x > \tag{4.19}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} a' \\ b' \end{pmatrix} = \begin{pmatrix} b' \\ a' \end{pmatrix} = - \begin{pmatrix} a' \\ b' \end{pmatrix} \tag{4.20}$$

식 (4.20)에 의해 식 (4.21)을 구한다.

$$a' = - b' \tag{4.21}$$

a'을 양의 실수라 한다면 식 (4.22)로 표현된다.

$$|\downarrow_x> = \frac{1}{\sqrt{2}}\{|\uparrow_z> - |\downarrow_z>\} \tag{4.22}$$

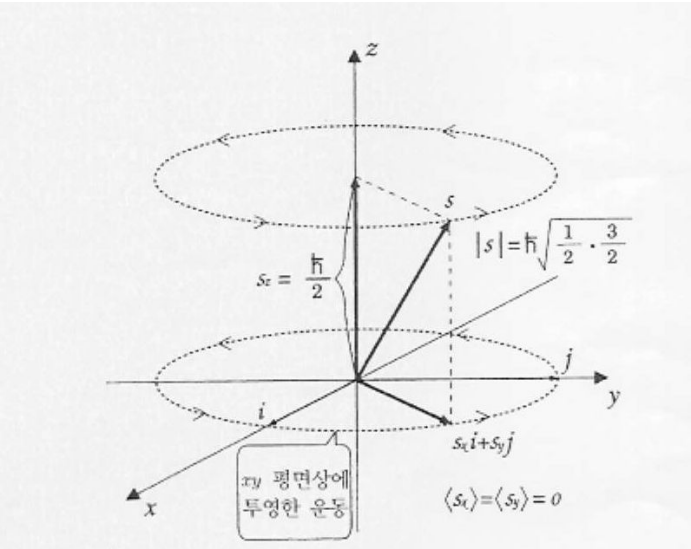
식 (4.22)는 직교조건 식 (4.23)도 만족하고 있음을 알 수 있다.

$$<\uparrow_x|\downarrow_x> = 0 \tag{4.23}$$

s_y가 갖는 $\pm\frac{1}{2}$의 고유상태도 같은 식으로 구할 수가 있으므로 식 (4.24)로 표현 가능하다.

$$|\uparrow_x>(|\downarrow_y>) = \frac{1}{\sqrt{2}}\{|\uparrow_z> + i|\downarrow_z>\}(\frac{1}{\sqrt{2}}\{|\uparrow_z> - i|\downarrow_z>\}) \tag{4.24}$$

양자컴퓨터에서는 양자비트(Qubit)라는 두 개의 양자상태를 선형결합 하여 사용한다. [그림 4.6]은 스핀 $\frac{1}{2}\hbar$인 입자의 상방 $|\uparrow>$상태를 벡터를 이용하여 설명하는 그림이며



[그림 4.6] 스핀 $\frac{1}{2}\hbar$인 입자의 상방 $|\uparrow>$상태

스핀 $\frac{1}{2}$인 입자를 양자비트로 나타내려면 0과 1을 식 (4.25)와 같이 스핀업과 다운 상태로 나타내고, 일반적인 양자비트는 식 (4.26)과 같이 나타낸다.

$$|0> = |\uparrow> = \begin{pmatrix}1\\0\end{pmatrix}, \quad |1> = |\downarrow> = \begin{pmatrix}0\\1\end{pmatrix} \tag{4.25}$$

$$|\psi> = a|0> + b|1> \tag{4.26}$$

1양자비트 상태의 n양자비트 확장식은 식 (4.27)과 같이 구해진다.

$$|\psi> = |\psi>_1 |\psi>_2 \cdots |\psi>_n = |1, 2, \cdots, n> \tag{4.27}$$

양자비트는 광자의 편광을 이용하여도 나타낼 수가 있으며 세로 $|\updownarrow>$, 가로 $|\leftrightarrow>$의 직선편광상태인

$$|0> = |\updownarrow>, \qquad |1> = |\leftrightarrow>$$

를 각각 0과 1에 대응시키면 된다. 두 개의 직선편광상태를 중첩시킨 상태인

$$|\psi> = a|\updownarrow> + b|\leftrightarrow> \tag{4.28}$$

식 (4.28)의 계수 a와 b를 적절하게 선택해 주면 $+45^\circ$와 -45°의 대각선 방향으로 직선편광된 식 (4.29)의

$$\begin{aligned} |\nearrow\swarrow> &= \frac{1}{\sqrt{2}}\{|\updownarrow> + |\leftrightarrow>\} \\ |\nwarrow\searrow> &= \frac{1}{\sqrt{2}}\{|\updownarrow> - |\leftrightarrow>\} \end{aligned} \tag{4.29}$$

광자상태나 우회전 원편광상태(시계방향) 식 (4.30)과

$$|R> = \frac{1}{\sqrt{2}}\{|\updownarrow> + i|\leftrightarrow>\} \tag{4.30}$$

좌회전 원편광상태(반시계방향) 식 (4.31)의

$$|L> = \frac{1}{\sqrt{2}}\{|\updownarrow> - i|\leftrightarrow>\} \tag{4.31}$$

광자상태를 만들 수 있다. 이들 상태를 이용하여도 역시 양자비트를 만들 수가 있다. 이 두 종류의 편광 중 가로·세로 직선편광상태는 식 (4.32)로 표현되며

$$\begin{aligned} |0>_{\oplus} &= |\updownarrow> \\ |1>_{\oplus} &= |\leftrightarrow> \end{aligned} \tag{4.32}$$

대각선편광상태는 식 (4.33)으로 가로·세로 직선편광상태의 중첩으로 바꿔 쓸 수 있다.

$$|0>_{\otimes} = \frac{1}{\sqrt{2}}\{|\updownarrow> + |\leftrightarrow>\}$$

$$|1>_{\otimes} = \frac{1}{\sqrt{2}}\{|\updownarrow> - |\leftrightarrow>\} \tag{4.33}$$

대각선편광상태를 가로·세로 직선편광 검출기를 이용하여 측정하며 $|\leftrightarrow>$와 $|\updownarrow>$ 상태가 $\frac{1}{2}$씩의 확률로 관측됨을 보여준다.

실난수 생성을 통한 양자 포톤 발생기에 의한 양자 회절판을 이용한 양자암호 전송 알고리즘은 추가적인 하드웨어적인 구성없이 기존의 광자 발생기를 이용하여 실난수를 생성하고 이를 스택 메모리 하나만으로 의사난수 사용을 없앨 수 있다는 장점이 있다.

V. 결 론

의사난수를 이용한 양자암호 키분배 방식은 의사난수의 기본 원칙들은 잘 적용하였으나 이들의 병합으로 인하여 확률 $\frac{1}{2}$인 상태들의 3가지 연산의 결합은 또 다른 난수성의 판별을 부르는 결과를 유도한다. 또한 두 개 이상 난수열에 의한 특이점들은 검산을 거친 후 사용 하여야 하고 의사난수의 기본성질들을 다 만족하더라도 각각의 난수열들이 피치 못하게 주기를 갖는 의사난수임을 감안하면 이진 비트열들이 2개 이상 얽혀서 나오는 경우는 모든 경우가 확률의 성질을 만족한다는 보장을 할 수가 없다. 이는 새로운 난수 테스트를 거쳐야 하고 실험 데이터가 충분히 만족한다 하더라도 컴퓨터 환경의 발달과 기술의 변화는 현 세대의 환경에서는 안전한 난수열들이더라도 근사점과 해법을 찾을 수 있게 된다.

추후 양자컴퓨터의 개발에 의한 난수의 주기성 파악은 예견되어 있는 상황에서 의사난수의 사용은 안정된 암호전송의 걸림돌일 수밖에 없다. 양자암호 시스템은 가장 안전한 암호형태인 원타임패드(One-time pad)를 이용한다는 점에서 역대 최대 안전성을 보전한다고 볼 수 있다. 하지만 이를 구현하기 위해서는 난수열의 사용이 필수불가결한 조건이고, 난수열 사용의 가장 간편한 방법이 기존에 나와 있는 안전성을 검증한 의사난수를 사용하는 것이다.

이는 기술 발달과 양자컴퓨터 환경의 도래로 인하여 안전성에서 거리가 멀어지고 있다. 이는 실난수의 사용으로 이어지는데 실난수 사용상의 가장 큰 문제점은 운영에 따른 비용이 너무 크다는 점으로 운영비용을 최소화 할 수 있다면 의사난수를 사용할 필요는 없어진다. 실난수는 자연계 내에서 존재하는 실측 난수들로 전자의 방출 각도, 빗방울의 떨어짐과 같은 다음 상태가 예측불가능하고 전위 상태에서 후의 상태의 변화에 대하여 독립적인 성질을 갖는다.

따라서 본 논문에서는 BB84 프로토콜 상에서 난수 병합에 따른 양자암호 취약성의 문제점을 해결하기 위하여 양자암호 시스템 상에서 의사난수를 생성하지 않는 실난수를 사용할 수 있는 더블·싱글 광자 발생기의 의사코드 알고리즘의 제안으로 양자암호 시스템에서 실난수의 효율성을 보장 받음을 확인 할 수 있었다. 또한 사용되는 벡터들과 이로 인한 실난수의 생성원리를 양자이론을 적용하여 양자 비트 발생의 난수성의 결과를 보여주었다.

특히, BB84 프로토콜 상에서 광자 발생기를 개선한 더블·싱글 광자 발생기의 차이점은 싱글 광자 발생기상에 데이터 흐름의 갭이 발생 되나 실난수를 구현하는데 있어서 비용적 요소와 하드웨어적인 불편함을 개선한 것으로 난수열 발생과정이 양자 비트를 적용한 것이라 할 수 있다. 그리고 기존 BB84 프로토콜 안에서 광자 발생기의 이중화를 통하여 ⊞, ⊠ 편광기를 서로 교차적으로 통과 시키면 마지막에 나오는 벡터들은 완전한 자연계의 난수들이 됨을 확인 할 수 있었다.

이를 스택에 일정한 양 만큼 보관하여 송신자의 편광기 선택과 랜덤 벡터 전송에 선택조건으로 사용되는 난수에 대처할 수 있다. 수신자도 마찬가지로 광자 발생기를 이용하여 실난수를 구현하고 수신편광기 선택에서 사용할 수 있다. 이러한 구조들은 모든 양자암호 시스템 내에서 난수열들의 사용에 있어 인간에 의한 취약성을 완전히 배제할 수 있는 근간이 될 것이라 생각한다.

향후 미래 컴퓨터 환경은 양자컴퓨터나 또는 현 시스템을 획기적으로 개선한 환경이 도래할 것이라 예측된다. 이에 양자암호 시스템은 차세대 암호시스템의 표준이 될 것 이며 이러한 상태에서는 의사난수의 효용가치는 점차 소멸하리라 본다. 또한, 광자 발생기를 이용한 실난수 생성과 이를 활용한 양자암호 시스템 개발은 차세대 컴퓨터 산업과 양자컴퓨터 시대의 기본 모형이 될 것이다. 많은 개발비와 사용상 비용이 많이 드는 일반적인 실난수의 사용은 상용화된 미래 양자컴퓨터 사회에서는 도태 될 것이고 산업 전반적인 분야에 광자 발생기나 또는 이와 유사한 형태의 알고리즘의 실현에 의해 암호전송 알고리즘 내에서 생성되는 실난수를 이용할 것이다.

하지만 단일광자를 발생하고 이를 제어할 수 있는 시스템은 완성되어있지 않는 상태여서 아직까지 광자다발을 이용한 암호전송이 보편화 되어 있다. 좀 더 안정적인 광자 발생기의 개발이 완성되는 시점이 멀지 않았으리라 짐작이 되며 그러한 시점에 본 논문에서 제안한 더블·싱글 광자 발생기를 적용한 의사코드 알고리즘은 여러 실난수 사용의 참고자료로 활용할 수 있을 것으로 사료된다.

[참고문헌]

[1] 원동호, 현대암호학, 그린, 2006.

[2] 진병문, 양자정보이론, 청범출판사, 2008.

[3] 황규범, 암호학의 이해, 경문사, 2009.

[4] 과학기술부, "The study on the quantum cryptography based on the quantum nature of the light", 1999.

[5] 한국정보보호센터, 표준전자서명용 의사난수 생성 알고리즘개발, 1999.

[6] 한국정보진흥원, "128비트 블록암호 알고리즘(SEED)개발 및 분석보고서", 2003.

[7] 김옥환, 최진석, 배상현, "암호화 된 사용자 보안 모듈을 적용한 분산 네트워크 보안 시스템 설계 및 구축", 조선대학교통계연구소, 제7권 1호, pp.121-132, 2005.12.

[8] 이성주, 최진석, "Random Number Statistical Test Using Fuzzy set Operation", Proceedings of KFIS Fall Conference 2002, p41, 2002.

[9] 임광철, 최진석, "이진 병합에 의한 양자암호 취약성", 한국지능시스템학회, 제20권 6호, pp.837-842, 2010.12.

[10] 최진석, 이성주, 김옥환, "유비쿼터스내에서 사용가능한 개인 인증방법", 조선대학교통계연구소, 제7권 1호, pp.185-193, 2005.12.

[11] 최진석, 이성주, "이진 알고리즘을 이용한 변형 시리얼테스트 설계에 관한 연구", 한국지능시스템학회, 제20권 1호, pp.76-80, 2010.02.

[12] ANSI X9.17(Revised), "American National Standard for Financial Institution Key Management(Wholesale)", American Bankers Association, 1985.

[13] C.H. Bennett, G. brassard, and S. Breidbart, "Quantum Cryptography Ⅱ How to Re-Use a One-time Pad Safely Even if P=NP", Nov 1982.

[14] G. Brassard, "Modern Cryptology: A Tutorial", Springer-Verlag, 1988.

[15] Chae Hoon.Lim "A Revised Version of CRYPTON", Information and Commuications Research Center.

[16] Charles H. Bennett, Gilles Brassard, Artur K. Ekert, "Quantum Cryptography", Scientific American, October 1992.

[17] J. Daemen, L.Knudsen and V. Rijmen, "The block cipher Square", In Fast

Software Encryption, Lecture Notes in Computer Science(LNCS) p267, Springer-Verlag, 1997.

[18] H. Feistel, "Block Cipher Cryptographic System", U.S. Patent #3,798,359,19 Mar 1974.

[19] N. Goots, B. Izotov, A. Moldovyan, N. Moldovyan, "Modern Cryptography: Protect Your Data with Fast Block Ciphers", A-LIST Publishing, 2003.

[20] L. R. Knudsen, "Block Ciphers-Analysis, Design and Applications", Ph.D Thesis, Computer Science department, Aarhus University, 1994.

[21] H. K. Lo and H. F. Chau, "Unconditional Security Of Quantum Key Distribution Over Arbitrarily Long Distances", Science, vol. 283, pp.2050-2056, 1999.

[22] Lokenath Debnath, Piort Mikusinski, "Introduction to Hilbert Spaces with Applications, Second Edition", Academic Press, 1999.

[23] A. Menezes, P. van Oorschot, S. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1996.

[24] NIST, "Instruction-level Parallelism in AES Candidates", Mar 1999.

[25] NIST, "Federal Information Processing Standards Publication 197- Specification for the Advanced Encryption Standard (AES)", Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, 2001.

[26] NIST, "Data Encryption Standard(DES)", Available: http://www.itl.nist.gov/fipspubs/fip46-2.htm, 1993.

[27] B. Schneier, "Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C (cloth)", 1996.

[28] D. Stinson, "Cryptography: Theory and Practice", CRC Press, 1995.

[29] M. Akkar and C. Giraud, "An Implementation of DES and AES, Secure against Some Attacks", CHES 2001, LNCS 2162, pp.309-318, 2001.

[30] C.H. Bennett, G.Brassard, S. Breidbart, and S. Weisner, "Quantum Cryptography, or Unforgeable Subway Tokens", Advances in Cryptology: Proceedings of Crypo 82, Plenum Press, pp.267-275, 1983.

[31] C.H. Bennett, G. Brassard, and A.K. J-M. Robert, "How to Reduce Your Enemy's Information", Advances in Cryptology-CRYPTO '85 Proceedings, Springer-Vergag, pp.468-476, 1986.

[32] C.H. Bennett, G. Brassard, and J-M. Robert, "Privacy Amplification by Public Discussion", SIAM Journal on Computing, vol. 17, no. 2, pp.210-229, Apr 1988.

[33] C.H. Bennett, G.Brassard, and A.K. Ekert, "Quantum Cryptography Without Bell's Theorem", Physical Review Letters, vol. 68, no. 5, pp.468-476, Feb 1992.

[34] C.H. Bennett, G. Brassard, C.C.repeau, and M.-H Skubiszewska, "Practical Quantum Oblivious Tansfer", Advances in Cryptology-CRYPTO '91 Proceedings, Springer -Verlag, pp.351-366, 1992.

[35] G. Brassard, "Quantum Cryptography : A Bibliography", SIGACT News, vol. 24, no. 3, pp.16-20, 1993.

[36] J. HÅstad, R. Impagliazzo, L.A. Levin, M. Luby, "A Pseudo random Generator from any One-way Function", SIAM J. Comput, vol. 28, pp.1364-1396, 1999.

[37] M Luby, C. Rack off, "How to construct pseudo random permutations from pseudo random functions", SIAM J. Comput, vol. 17, pp.373-386, 1988.

[38] D. Mayers, "Quantum key distribution and string oblivious transfer in noisy channels", in Advances in Cryptography-Proceedings of Crypto'96, Lecture Notes in Computer Science, vol. 1109, pp.343-357, edited by N. Koblitz, Springer-Verlag, New York, 1996.

[39] R.A Rueppel, "On the Security of Schnorr's Pseudo-Random Sequence Generator", Advances in Cryptology-EUROCRYPT,89 Proceedings, pp.423-428, 1990.

[40] B. Schneier, "The uses and Abuses of Biometrics", Communications of the ACM, vol. 42, no. 8, p136, 1999.

[41] C. E. Shannon, "Communication theory of secrecy system", BSTJ, vol. 28, pp.656 -715, 1949.

[42] E.H. Sibley, "Random Number Generators : Good Ones Are Hard to Find," Communications of the ACM, vol. 31, no. 10, pp.1192-1201, 1988.

[43] Whitfield Diffie and Martin E. Hellman. "New directions in cryptography", IEEE Transactions on Information Theory, vol. 22, no. 6, pp.644-645, November 1976.