기술기준연구	99-2

블록암호알고리즘 안전성 평가항목 분석

A Study on the Security Evaluation Factors of Block Ciphers

1999. 12.

序 文

정보통신 기술의 발전과 함께 정보화 사회가 고도화 되어감에 따라 정보보호의 중요성이 날로 증대하고 있습니다. 정보보호는 이제 공공기관뿐 아니라 민간기관과 개인에게 있어서도 중요한 문제가 되고 있습니다. 정보보호의 대상은 크게 컴퓨터에 저장된 개인 및 기관의 정보와 인터넷과 같은 네트웍상의 정보가 있습니다. 이러한 정보를 올바르게 보호하기 위해서는 많은 정보보호 관련 기술이 필요하고 현재 다양한 분야에서 연구가 진행되고 있습니다.

정보보호의 다양한. 기술 중에 가장 기본적인 것이 암호기술입니다. 암호기술은 이제 군, 정보기관 등의 한정된 분야에서 정보통신의 모든 분야로 확대되어 사용되고 있습니다. 다양한 정보보호제품에도 암호기술이 사용되고 있습니다. 이에 따라, 정보보호제품의 평가기준과 평가를 담당하고 있는 한국정보보호센터에서는 암호모듈에 대한 평가를 준비하기 위해 가장 많이 사용되고 있는 블록암호알고리즘의 안전성 평가항목 분석 연구를 일차로 진행하였습니다.

본 보고서에서는 블록암호알고리즘의 기본 개념, 설계기준, 기본구조를 정리하여 소개하였습니다. 또한, 블록암호알고리즘 안전성 평가를 위해 먼저 안전성평가항목을 분류하고, 각 분석항목에 대한 특성과 평가조건 등을 설명하였습니다. 최종적으로 분석된 항목들을 바탕으로 정보통신망 침입차단시스템 암호모듈평가에 적용 가능한 평가항목을 도출하였습니다. 본 보고서에서 정리된 평가항목은 기본적인 사항으로 암호모듈을 이용한 정보보호제품 개발자와 암호모듈을 사용한 정보보호제품의 도입을 고려하고 있는 사용자에게도 유용한 정보가 될 것으로 확신합니다.

향후 지속적인 보완작업과 연구를 통해 암호모듈 및 제품을 평가할 수 있는 평가기준과 평가기술을 확립할 것이며, 본 보고서의 발간을 위해 노력한 연구원들의 노고에 감사드립니다.

1999年 12月

한국정보보호센터 원장

이 철 수

제 출 문

본 연구보고서는 “정보보호 평가기준 개발”의 일환으로 정보화촉진기본법 개정안 기반의 침입차단시스템 평가수행 Task Force Team에서 수행한 블록암호알고리즘 안전성 평가 항목 분석결과 보고서입니다.

1999년 12월

연구 책임자 : 기획평가부장 고승철
참여 연구원 : 기술기준팀장 이경구
기반기술팀장 박성준
선임 연구원 김학범
연 구 원 이인수
연 구 원 조규민
외부 전문가 : 배재대 교수 성수학
한양대 교수 송정환
고려대 교수 이상진

요 약 문

1. 제목 : 블록암호알고리즘 안전성 평가항목 분석

2. 연구개발의 목적 및 중요성

암호제품 및 정보보호시스템에 포함된 암호모듈의 블록암호알고리즘 안정성을 평가하기 위한 평가항목을 도출하고 향 후 블록암호알고리즘 안전성 분석 모델 개발의 기초자료를 작성하고자 한다.

3. 연구개발의 내용 및 범위

본 연구에서는 암호기술 중 대칭키 블록 암호알고리즘(통상 블록암호알고리즘이라 칭함)을 다루고자 한다. 현재 암호알고리즘은 통상 안전성과 효율성 측면을 모두 고려하여 요구되는 적합한 안전도를 지원할 수 있도록 최적 설계된다. 따라서, 본 연구에서는 알고리즘의 구조적 취약성에 기반을 둔 다양한 분석기법들을 분석하고, 기존의 블록암호알고리즘의 안전성 분석 항목의 특성을 분석하여 안전성 평가항목을 분류・도출한다.

4. 연구결과

본 연구에서는 블록암호알고리즘의 기본 개념, 설계기준, 기본구조를 정리하였다. 또한, 블록암호알고리즘 안전성 평가를 위해 먼저 안전성 평가항목을 분류하고, 각 분석항목에 대한 특성과 평가조건 등을 설명하였다. 최종적으로 분석된 항목들을 바탕으로 정보통신망 침입차단시스템 암호모듈 평가에 적용 가능한 평가항목을 도출하였다.

5. 활용에 대한 건의

본 고에서는 기존에 알려진 암호알고리즘 안전성 평가항목을 분석 · 정리하였다. 본 고에 나열된 평가항목을 모두 만족하는 알고리즘이라도 안전한 것은 아니지만, 본 평가항목을 통과한 암호알고리즘은 현재까지의 공격에 안전하다는 것을 의미한다. 정리된 각 항목은 안전도를 검증하기 위한 최소 통과 기준이며, 암호알고리즘의 분석 및 평가의 기본자료로 활용 가능하다. 향후, 암호 평가를 수행하기 위한 평가모델의 기반구축에 활용될 수 있다.

6. 기대효과

본 연구개발 결과는 암호기능을 위해 가장 많이 사용되는 블록암호알고리즘의 안전성을 평가하기 위한 기본 항목을 도출한 것으로 암호 안전성 평가모델로 발전할 수 있다. 본 고에서 정리된 평가항목은 암호기능의 개발을 고려하는 개발자와 암호기능을 포함한 제품을 도입하려 사용자에게 블록암호알고리즘의 안전성을 확인하기 위한 기본항목으로 제시되어 정보보호제품의 활성화에 기여할 수 있을 것이다.

SUMMARY

1. Subject : Study on the Security Evaluation Factors of Block Ciphers

2. Objective and Importance of Research and Development

Study on the Security Evaluation Factors of Block Ciphers included in the cipher machines and IT products. It will also be used as an input for the development of block cipher security model.

3. Scope of the R & D

The R&D covers secret key block cipher, which is usually called block cipher. Block cipher is designed most effectively considering both security and efficiency. This R&D analyzes analytic methodologies based on algorithm structures. It will also develop security evaluation factors by analyzing security analytic factors of block cipher.

4. Results of the R & D

As a result of the R & D, it defines basic concepts, basic architecture, design criteria of the block ciphers. Also, it defines security concept and security analyzation factors for analyzing block ciphers. Futhermore, it defines characteristics and conditions for each security analyzation factors. Finally, defines security evaluation factors for block ciphers.

5. Practical Use

This paper defines block cipher security evaluation factors and analyzed results. However, even though an algorithm satisfies a1l the block cipher security evaluation factors, it does not mean that an algorithm is secure. What it means is that it is safe against all the known vulnerabilities up till now. Each factors are the minimum requirements to pass in security wise and it can be used as a basic data to evaluate and analyze block ciphers. It will also be used as an input for developing evaluation system for block cipher.

6. Expected Benefits

The result of this report is to draw up the security evaluation factors of block cipher which is most often used to implement to provide encryption functionality. And this could be used to develop cipher security evaluation model. Security evaluation factors will enhance the use of information security products.

목 차

다. 차분 분석

1) 기본 차분 분석
2) 조건부 차분 분석
3) 부정 차분 분석
4) 불능 차분 분석
5) 고계 차분 분석
6) 부메랑 공격

라. 선형근사 분석

1) 기본 선형근사 분석
2) Davies 공격
3) 다중 선형근사 분석
4) 선형/비선형 근사 분석
5) 일반 선형근사 분석
6) 분할 공격 / Modn 분석

마. 기타

1) 보간 다항식 공격
2) 차분선형근사 분석

바. 난수 통계특성 검정법

1) 도수 검정법
2) 도수-m 검정법
3) Binary derivative 검정법
4) Poker 검정법
5) Runs 검정법
6) 계열(Serial) 검정법
7) 자기상관(Autocorrelation) 검정법
8) Change Point 검정법
9) 충돌(Collision) 검정법
10) Cuppon collector's 검정법
11) Gap 검정법
12) 순열(Permutation) 검정법
13) 최대값(Maximum) 검정법
15) 선형 복잡도(Linear Complexity) 검정법
16) 선형 복잡도 프로파일(Linear Complexity Profile) 검정법
17) Zip-Lempel 복잡도 검정법
18) Walsh-Power Spectrum 검정법
19) 새로운 일반적 검정(New Universal Test)

사. 부가정보 이용 공격

1) DPA(Differential Power Analysis)
2) DFA(Differtial Fault Analysis)
3) Timing 공격en

제 5 장 결 론

[첨부 1] 주요 旣 개발 블록 암호알고리즘 명세

[첨부 2] 주요 참고 문헌

Contents

5) Slide attack

4.3 Differential cryptanalysis

1) DC
2) Conditional DC
3) Truncated DC
4) Impossible DC
5) Higher-Order DC
6) Boomerang attack

4.4 Linear cryptanalysis

1) LC
2) Davies's attack
3) Multiple linear approximation

4) Linear/Non-linear approximation
5) Generalized LC
6) Partitioning Cryptanalysis / Mod n Cryptanalysis

4.5 Others

1) Interpolation attack
2) Differential Linear cryptanalysis

4.6 Random Tests

1) Frequency Test
2) Frequency-m Test
3) Binary derivative Test
4) Poker Test
5) Runs Test
6) Serial Test
7) Autocorrelation Test
8) Change Point Test
9) Collision Test
10) Cuppon collector's Test
11) Gap Test
12) Permutation Test
13) Maximum Test
15) Linear Complexity Test
16) Linear Complexity Profile Test
17) Zip-Lempel Complexity Test
18) Walsh-Power Spectrum Test
19) New Universal Test

제 1 장 개 요

20세기 후반 급속한 컴퓨터 및 네트워크의 발달은 정보 관리 및 유통방식에 커다란 변혁을 불러 일으켰다. 즉, 종이와 같은 기존의 전통적 수단을 통하여 관리되던 정보가 컴퓨터 등 새로운 저장 매체를 통해 디지털화되어 관리됨으로써 정보 관리비용의 절감과 유통면에서의 효율성이 증대되었다. 산업사회에서의 고비용 저효율적 정보관리 및 유통방식이 지식정보사회에서의 저비용 고효율적인 방식으로 급속히 대체되고 있는 현 상황은 새로운 속성의 정보 특성에 따른 새로운 방식의 정보 관리 방식을 나타내는 것이다. 이러한 정보관리 기술 중의 하나가 바로 정보보호 기술이며 정보보호 기술이 제공하는 서비스 중 기밀성 서비스는 대표적 암호서비스 중의 하나이다. 과거 암호기술은 기밀성 서비스 측면이 강조되는 군사 · 의교 등 국가분야에 국한되어 개발 · 활용되었으나, 현재 그 수요는 전자상거래 등 민간분야로 확대 · 다양화되어 다양한 기술 및 서비스가 연구 · 개발되고 있다.

본 연구에서는 이러한 암호기술 중 대칭키 블록 암호알고리즘(통상 블록 암호알고리즘이라 칭함)을 다루고자 한다. 대칭키 블록 암호알고리즘은 기밀성(confidentiality) 기능을 제공할 수 있는 알고리즘으로[1], 안전도 중심의 전통적인 기밀성 서비스 제공 기술이 상업 용도로 변모를 거치면서 효율성 측면이 강조되어 개발된 기술이라 할 수 있다.

현재 암호알고리즘은 통상 안전성과 효율성 측면을 모두 고려하여 요구되는 적합한 안전도를 지원할 수 있도록 최적 설계된다. 우선, 가까운 미래(암호알고리즘 예상 사용기간)의 계산능력 수준을 예측하여 지원 안전도 및 암호알고리즘의 사용주기(life cycle)를 결정한다. 특히, 암호알고리즘의 안전도는 계산능력 뿐만 아니라, 알고리즘의 구조적 취약성에 기반을 둔 다양한 분석기법들(기존의 분석기법 또는 설계시 예상하지 못한 분석기법)과 밀접한 관계가 있다[2].

---

1) 블록 암호알고리즘은 의사난수생성기(Pseudo Random Generator), 스트림 암호알고리즘(Stream Cipher), 메시지 인증 코드(MAC, Massage Authentication Code) 및 해쉬함수(Hash Function)를 설계하는 하는 데에 활용될 수도 있다.

2) 암호알고리즘 설계시 효율성에 지나치게 치중할 경우, 원래 암호알고리즘의 사용 목적을 달성하기 어렵고, 안전도에 치중할 경우, 과다 비용 지출이라는 경제적 손실을 끼칠 수 있다. 안전한 암호알고리즘을 설계하거나 효율적인 암호알고리즘을 설계하는 것은 그리 어렵지 않다. 그러나 안전하고 효율적인 암호알고리즘을 개발하는 것은 무척 어려운 일로 알려져 있다.

따라서 암호알고리즘의 사용여부를 판단하기 위해서는 주기적인 재 검증과정이 필수적으로 요구된다.

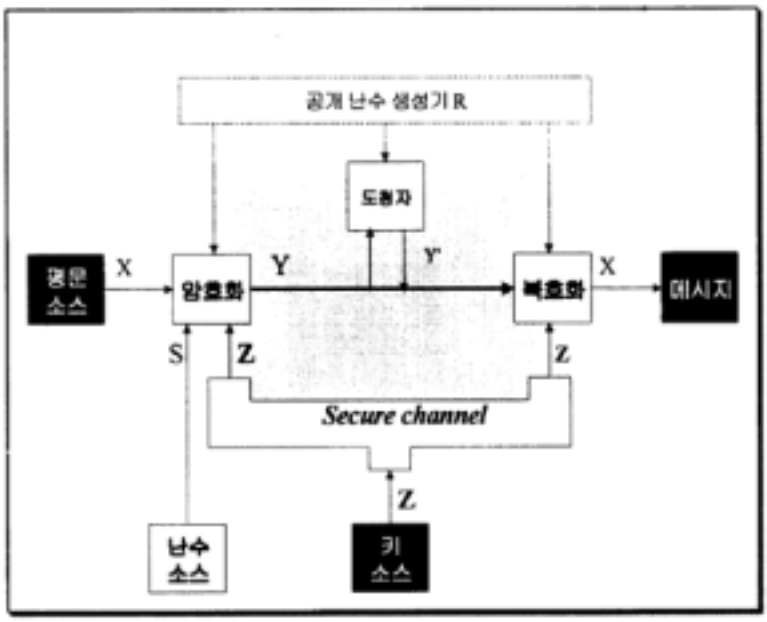
본 연구에서는 위와 같은 특성을 갖는 기존의 블록 암호알고리즘의 안전성 분석 항목의 특성을 분석하여 안전성 평가항목을 분류 · 도출한다. 또한 향후, 블록 암호알고리즘 안전성 분석 모델 개발의 기초 자료로 활용하고자 한다.

제 2 장 블록 암호알고리즘

가. 정의

1) 암호알고리즘의 정의 및 분류

암호알고리즘(Cryptographic Algorithm, Cipher)이란 암호화(encryption) 및 복호화(decryption)에 사용되는 수학적 함수를 말하고, 키 생성 알고리즘, 암호화 알고리즘 및 복호화 알고리즘으로 구성된다.

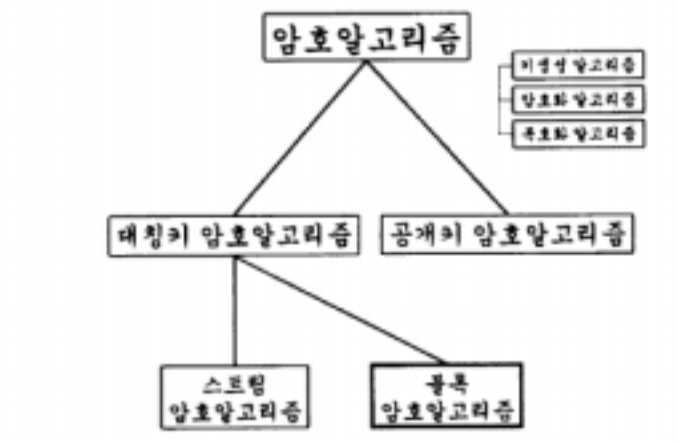


일반적으로 암호알고리즘은 키의 형태에 따라 공개키 암호알고리즘(Public Key Algorithm 또는 Asymmetric Algorithm)[3]과 대칭키 암호알고리즘(Symmetric Algorithm)[4]으로 대별되며, 대칭키 암호알고리즘(또는 관용키 암호알고리즘)은 메시지 처리 크기(단위)에 따라 스트림 암호알고리즘(Stream Cipher)과 블록 암호알고리즘(Block Cipher)으로 나누어 볼 수 있다.

---

3) 공개키 암호알고리즘의 개념은 1976년 W.Diffie와 M.E.Hellman이 처음 소개하였다. 공개키 암호알고리즘은 개인키(private key, 비밀 보관)와 공개키(public key, 공개)라는 두 개의 키 쌍을 사용하며, 개인키와 공개키는 서로 연계되어 있으나, 공개키로부터 개인키를 추출하는 것이 계산상 불가능하다는 가정하에 설계된다.

4) 비밀키 암호알고리즘(secret-key cryptography)이라고도 불리며, 암호화와 복호화에 사용되는 키가 같은 알고리즘을 말한다.



스트림 암호알고리즘은 한 번에 평문(또는 암호문)의 한 비트(또는 한 바이트) 단위로 연산(암호화 또는 복호화)을 수행하며, 블록 암호알고리즘은 고정된 크기(보통 64 또는 128비트)의 블록단위로 연산을 수행한다.

2) 블록 암호알고리즘의 정의

블록 암호알고리즘이란 n비트 평문 블록을 n비트 암호문 블록으로 바꾸는 함수를 말한다(n:블록 크기). 즉, n비트 블록 암호알고리즘이란 각 키 $k \in K$에 대하여, $E(P,k)$ (= $E_k(P)$, 암호화 함수)가 V_n에서 V_n으로 가는 역 변환 ($D_k(C) = D_k(E_k(P)) = P$, 복호화함수), 이 가능한 함수 E: $V_n \times K \rightarrow V_n$를 말한다. 블록 암호알고리즘에서 중요한 두 매개변수는 키 길이와 블록 크기이다. 참고로 현재 권고되는 키 길이는 최소 75~90 비트이고, 주로 사용되는 블록 크기는 64 비트 또는 128 비트이다[5].

비용 대비 해독에 걸리는 시간						
	키 길이(비트)					
비용	40	56	64	80	112	128
$100 K	2 초	35 시간	1 년	70,000 년	10^{14} 년	10^{19} 년
1 M	.2 초	3.5시간	37 일	7000 년	10^{13} 년	10^{18} 년
10 M	.02 초		4 일	700 년	10^{12} 년	10^{17} 년
100 M	2 ms	13 초		70 년	10^{11} 년	10^{16} 년
1 B	.2 ms	1 초		7 년	10^{10} 년	10^{15} 년
10 B	.02 ms	.1 초		245 일	10^{9} 년	10^{14} 년
100 B	2 μs	.01 초	32 초		10^{8} 년	10^{13} 년
1 T	.2 μs	1 ms	3 초	2.4 일	10^{7} 년	10^{12} 년

※ 2000~2005년까지 현황(Gartner社 1999년)

5) 현재 계산 능력을 기반으로 해독이 불가능하도록 결정된다

나. 설계 기준

1) Shannon의 기준

현대 암호 이론은 정보 이론(information theory)의 기본적 개념 없이는 전개가 불가능하다. 이 분야의 始祖인 Claude E. Shannon은 1948년[6] noisy 채널을 통한 통신과 관련된 공식화된 원리를 발표하였다. 그 후 일년 뒤[7], 정보 이론을 보안 시스템과 연계하여 발전시키는 등, 현대 암호의 이론적 기반을 제공하였다. Shannon은 상기 논문에서 복잡하고 안전한 암호시스템을 설계하기 위하여 단순하고 안전하지 않은 두 개의 다른 종류 암호알고리즘(혼동[8]과 확산[9])을 결합하는 방법(product cipher)[10]을 제안하였다. 더욱이 암호시스템을 평가하기 위한 5가지 기준을 다음과 같이 제시하였다.

o 안전도 량(Amount of secrecy)
- 완전 안전도(perfect secrecy)
o 키 크기(Size of key)
- 키는 반드시 안전한 통로를 통하여 전달되어야 한다
- 가능하면 짧을수록 좋다
o 암호화/복호화 연산의 복잡도(Complexity of enciphering & deciphering operations)
- 암호화와 복호화 연산은 단순할수록 좋다.
o 에러 전파(Propagation of errors)
- 에러는 최소화될수록 좋다
o 메시지 확장(Expansion of message)

---

6) "A mathematical theory of communication" Bell Sys. Tech. J., 27:623-656, 1948
7) "Communication theory of secrecy systems" Bell Sys. Tech. J., 27:627-715, 1949
8) 혼동(confusion)은 암호문과 키 사이의 관계를 매우 복잡하게 만드는 것을 말한다.
9) 확산(diffusion)은 잉여정보가 긴 영역에 골고루 퍼지도록 평문의 통계적 구조를 만드는 것을 말한다. 평문과 암호문사이의 관계를 복잡하게 만드는 것이다.
10) 현재 블록 암호알고리즘은 대부분 이 기준을 기반으로 설계되고 있다.

- 암호화 과정에서 메시지의 크기가 증가되면 좋지 않다.

2) Massey의 기준

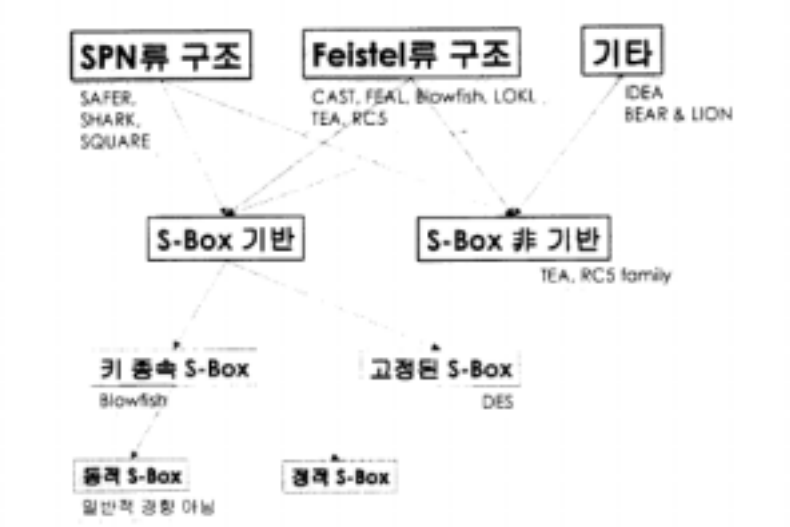
o 혼동(Confusion) : 대치(Substitution)
- 암호문의 통계적 특성을 암호분석가(해독가)가 알 수 없도록 평문의 통계적 특성에 복잡하게 의존시킨다
o 확산(Diffusion) : 치환(Permutation)
- 평문의 각 디지트와 암호키의 각 디지트가 암호문의 많은 디지트에 영향을 미친다
o 충분히 큰 블록, 키 크기
o 알려진 공격에 대한 저항
- 차분 분석법(Differential attacks)
- 선형근사 분석법(Linear attacks)
o 모든 키들이 동등하게 적합하다.
o 관계가 단순하지 않다.
o 높은 비선형 위수를 가진다.

3) 기타 설계 기준들

o 안전도
- 보호될 정보의 가치와 관련된 안전도
o 효율성
- H/W 또는 S/W에서의 효율성
o 비용
- 설계, 특허, 구현 그리고 키관리의 비용
o 가용성
- 상업적 목적과 국가적 목적에 따른 적합성
※ 강한 암호알고리즘이 수출 통제되기도 함
o 표준화

- 장비 재사용의 최대화 및 어떤 보안 특성을 제공하기 위해
o 에러 전파와 동기화
- 통신 프로토콜의 하위 계층에서의 암호화를 위해서는 매우 중요함

다. 기본 구조

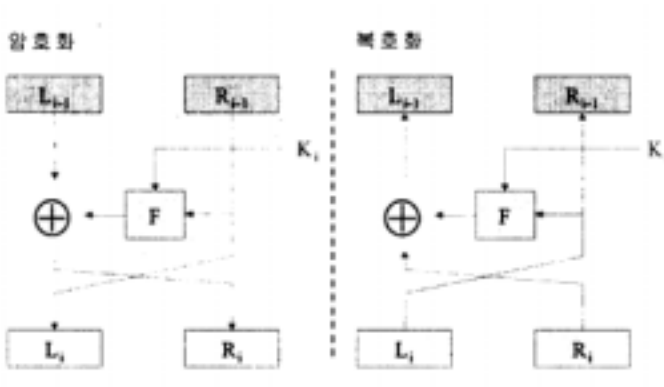


1) Feistel 구조

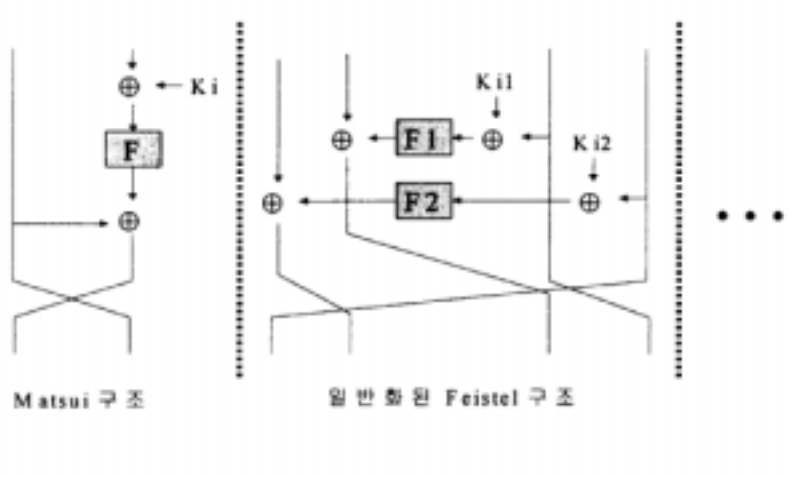
Feistel 구조란 각 t비트인 L_0, R_0블록으로 이루어진 2t비트 평문블록(L_0, R_0)을 r라운드(r≥1)를 거쳐 암호문 (L_r, R_r)을 출력하는 반복 구조를 말한다. 반복 구조란 평문 블록이 몇 번의 라운드를 거쳐 암호화를 수행하는 것을 말하고, 라운드 i(1≤i≤r)란 암호키 K로 부터 유도된 각 서브키 K_i(또는, 라운드 키라 불림)를 중요 입력으로 하는 $L_i = R_{i-1}$, $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$를 통해 $\{L_{i-1}, R_{i-1}) \xrightarrow{K_i} (L_i, R_i)$로 바꾸어 주는 함수를 말한다. 또한, 전체 알고리즘의 라운드 수는 요구되는 보안 강도와 수행 효율성의 상호 절충적 관계에서 결정된다.

보통 Feistel 구조는 3라운드 이상이며, 짝수 라운드로 구성된다. 이러한 Feistel 구조는 ①라운드 함수에 관계없이 역변환이 가능하며(즉, 암 · 복호화 과정이 같음). ②두 번의 수행으로 블록간의 완전한 확산(Diffusion)이 이루어지며, ③알고리즘의 수행속도가 빠르고, ④H/W 및 S/W 구현이 용이하고, ⑤아직 구조상의 문제점이 발견되고 있지 않다는 장점을 지니고 있다.

(1) 기본 Feistel 구조



(2) 수정된 Feistel 구조



2) SPN(Substitution-Permutation Network) 구조

SPN 구조란 대치(substitution, 암호키 사용)와 치환(Permutation, 암호키 사용하지 않음)을 반복적으로 사용하는 구조이다. 즉, S-box라 불리는 작은 부분 블록의 비선형 대치와 비트 치환을 연계하여 반복하는 구조이다.

SPN 구조는 일반적으로 다음의 성질을 만족하여야 한다

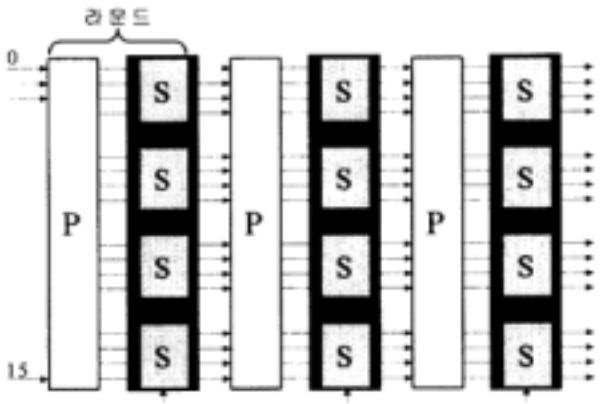
o 쇄도 특성(Avalanche Property)

- 입력의 한 비트가 변화될 때, 출력 비트의 약 절반이 변화되어야 한다.

o 완비 특성(Completeness Property)

- 출력의 한 비트는 입력의 모든 비트로부터 영향을 받아야 하고 이것은 가능한 모든 키 값에서 만족하여야 한다.

특히, SPN 구조는 일반적으로 암호화 함수와 복호화 함수가 달라 H/W와 S/W로 구현 시 비용이 많이 든다는 단점이 있다.



제 3 장 블록 암호알고리즘 안전성 분석 항목 분류

가. 안전성 이란?

암호알고리즘은 수 천년 전부터 개발 · 사용되어 왔다. 그러나 고대암호는 현대 컴퓨터 기술의 진보로 인하여 더 이상 안전을 보장받을 수 없게 되었다. 따라서 요구되는 안전도를 지원하기 위한 암호기술도 그와 발맞추어 발전하고 있다. 과거 암호알고리즘은 비공개를 전제로 설계·관리되었다. 즉, 암호알고리즘이 노출될 경우 쉽게 해독 가능하여 암호알고리즘 자체의 보호가 중요 보안 매개변수이었다. 그러나 암호알고리즘 보호에 대한 위협이 매우 크고[11], 또한 다수가 사용하기에는 불편한 점이 많은 비실용적 접근방식이었다. 따라서 현재 암호알고리즘 설계시에는 알고리즘을 공개하고[12] 그 대신 암호키를 보안 매개변수로 가정한다.

1) 이론적 안전도(정보이론적 관점)

1949년 Shannon은 정보이론에 기반을 둔 완전 안전도(Perfec Secrecy)를 정의[13]하였다. 즉, 완전한 암호알고리즘(perfect cipher)은 암호문 단독 공격(COA)에 unconditionally secure[14]하다. 그 예로 Shannon은 Vernam 암호알고리 즘(OTP, One-Time Password)이 완전한 암호알고리즘임을 보였다.

11) 예: RC2 및 RC4는 초기에 비공개, 그러나 바로 인터넷 등을 통하여(alleged)알고리즘이 알려 짐

12) Kerckoff's principle (19C 독일 암호학자) : 공격자는 비밀키를 제외한 모든 암호화 및 복호화 과정을 (자세히) 알고있다

13) 모든 평문 P와 모든 암호문 C에 대하여 Pr(P)=Pr(P | C)가 성립하면, 비밀키 암호알고리즘이 완전하다(perfect)고 한다. 즉, 암호문으로부터 평문의 어떠한 정보도 얻을 수 없다.

14) 무한한 계산 능력을 가진 공격자도 해독할 수 없다.

그러나 완전한 암호알고리즘은 키 길이가 메시지 길이보다 길거나 같아야 하는 비실용적 암호알고리즘[15]이라 할 수 있다. 따라서 실용적 암호알고리즘[16]의 개발 필요성이 대두되었고, 계산복잡도 이론에 기반을 둔 보다 실용적 암호알고리즘개발이 시작되었다.

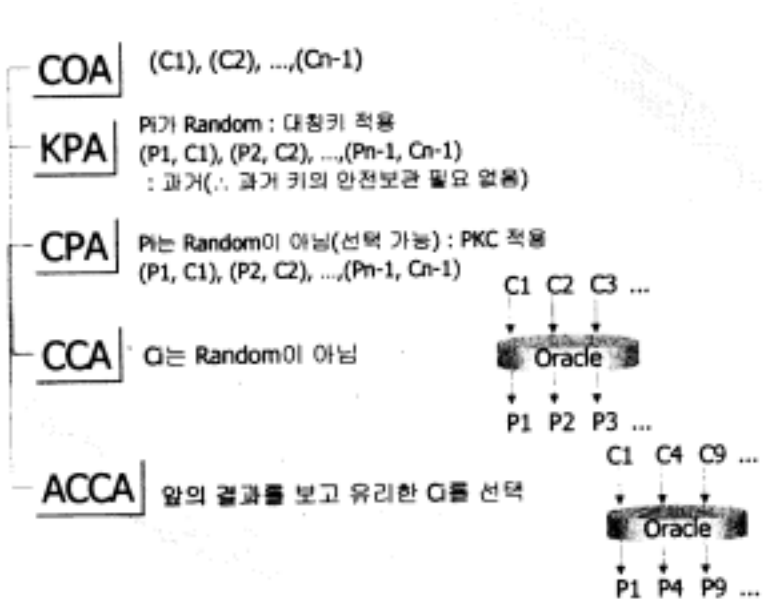
2) 실용적 안전도(계산복잡도 관점)

기본적으로 계산복잡도에 기반을 둔 암호알고리즘은 공격자의 계산 능력이 무한하다면, 암호문으로부터 평문의 정보를 추출하거나 키를 찾을 수 있다. 그러나 유한한 계산 능력을 가진 공격자는 그 정보를 추출하거나, 혹은 추출된 정보로부터 평문 또는 키를 유추할 수 없다. 본 연구에서 다루는 블록 암호알고리즘도 바로 계산복잡도 기반의 안전도를 제공하는 알고리즘이다. 즉, 블록 크기와 키 크기가 유한하게 고정됨으로써 키 전수검색이 가능하기는 하지만, 그것이 현실적(유한 계산 능력)으로 불가능하다는 사실에 기반을 둔다. 즉, 키 전수검색보다 적은 키 공간에서 키를 찾을 수 있는 방법을 찾는 것이 암호분석이며, 이 경우 그 암호알고리즘은 원래의 안전도를 지원하지 못한다고 한다.

15) 그러나 외교관이나 간첩 등은 여전히 사용하고 있기도 하다. 또한 과거 모스크바와 워싱턴간의 red telephone에서 사용되기도 하였다.
16) 공격자가 유한한 계산능력을 가지고 있다는 가정 하에서 설계한다. 즉, 공격자는 확률론적 알고리즘을 다항식 시간 안에 돌릴 수 있다.

나. 안전성 분석이란?

1) 공격방식 분류



가) Ciphertext-only attack(COA, 암호문 단독 공격)

해독자가 같은 알고리즘에 의해 생성된 암호문을 가진다고 가정할 때, 평문 또는 키를 추론하는 공격법이다.

알려진 정보 : $C_1 = E_K(P_1), C_2 = E_K(P_2), \cdots C_i = E_K(P_i)$

추론하는 것 : $P_1, P_2, \cdots P_i, K$또는 $C_{i+1} = E_K(P_{i+1})$로부터 P_{i+1}

나) Known-plaintext attack(KPA, 기지 평문 공격)

해독자가 암호문뿐만 아니라 대응되는 평문을 안다고 가정할 때, 키 또는 평문을 추론하는 공격법이다.

알려진 정보 : $P_1, C_1 = E_K(P_1), P_2, C_2 = E_K(P_2), \cdots P_i, C_i = E_K(P_i)$

추론하는 것 : K또는 $C_{i+1} = E_K(P_{i+1})$로부터 P_{i+1}

다) Chosen-plaintext attack(CPA, 선택 평문 공격)

해독자가 암호문과 대응되는 평문뿐만 아니라 원하는 평문을 가진다고 가정할 때, 키 또는 평문을 추론하는 공격법이다.

알려진 정보 : $P_1, C_1 = E_K(P_1), P_2, C_2 = E_K(P_2), \cdots P_i, C_i = E_K(P_i)$

단, 해독자가 평문 $P_1, P_2, \cdots, P_i$를 선택

추론하는 것 : K또는 $C_{i+1} = E_K(P_{i+1})$로부터 P_{i+1}

라) Adaptive-chosen-plaintext attack(ACPA)

해독자가 선택한 평문과 대응되는 암호문뿐만 아니라 암호문에 따라 평문을 선택할 수 있다고 가정할 때, 키 또는 평문을 추론하는 공격법이다. Chosen-plaintext attack 에서는 해독자가 많은 평문을 선택하지만, adaptive-chosen-plaintext attack 에서는 해독자가 적은 량의 평문을 선택하며 대응되는 암호문의 결과에 따라 평문을 선택한다.

마) Chosen-ciphertext attack(CCA, 선택 암호문 공격)

해독자가 선택한 암호문과 대응되는 평문을 안다고 가정할 때, 키 또는 평문을 추론하는 공격법이다.

알려진 정보 : $C_1, P_1 = D_K(C_1), C_2, P_2 = D_K(C_2), \cdots C_i, P_i = D_K(C_i)$

단, 해독자가 암호문 $C_1, C_2, \cdots, C_i$를 선택

추론하는 것 : K

바) Chosen-text attack

Chosen-plaintext attack과 chosen-ciphertext attack을 chosen-text attack이 라고 한다.

사) Chosen-key attack(선택 키 공격)

해독자가 키를 선택할 수 있다는 것이 아니라 다른 키들 사이의 관계를 알고 있다고 가정할 때, 키를 추론하는 공격법이다.

아) Rubber-hose cryptanalysis

해독자가 위협, 갈취, 고문하여 키를 획득하는 방법이다.

2) 블록 암호알고리즘에서의 안전성 분석

암호알고리즘이 안전하기 위해서는 암호문으로부터 평문 또는 키에 관한 어떠한 정보도 이끌어낼 수 없어야 하며, 더 나아가서 이전에 사용되었던 임의의 평문, 암호문, 키를 알고 있다고 하여도 현재의 암호문에 대응되는 평문 또는 키에 관한 정보를 이끌어 낼 수 없어야 한다. 이러한 관점에서 블록 암호알고리즘도 안전해야 한다.

블록 암호알고리즘의 안전성은 다음 세 가지 문제를 고려하여야 한다.

① 평문과 암호문이 주어졌을 때 키를 찾을 수 있는가?

② 주어진 평문, 암호문으로부터 미지의 암호문에 대한 평문을 구할 수 있는가?

③ 서로 다른 키로 암호화된 평문과 암호문이 있을 때 키를 구할 수 있는가?

이러한 문제에 안전하기 위해서 블록 암호알고리즘은 다음을 만족하여야 한다.

① 평문과 암호문은 키를 모르는 상태에서 어떠한 연관관계도 이끌어 낼 수 없어야 한다.
② 키와 암호문 사이에도 연관 관계를 찾을 수 없어야 한다.
사실 알고리즘에 의해 암호문이 생성되었기 때문에 평문과 암호문, 키와 암호문 사이에는 특별한 연관 관계가 존재할 수밖에 없다. 따라서 키 전수 조사보다 적은 계산량으로는 미지의 암호문으로부터 키나 평문의 어떠한 정보도 이끌어 낼 수 없을 때 주어진 블록 암호알고리즘은 안전하다고 말할 수 있다. 또는 미지의 암호문으로부터 키나 평문의 정보를 이끌어 내는데 필요한 계산량을 주어진 블록 암호알고리즘의 안전도라고 말할 수 있다.
블록 암호알고리즘의 대표적인 공격법인 차분 분석법은 평문 쌍과 대응되는 암호문 쌍으로부터 키에 대한 정보를 추출하는 선택 평문 공격이고, 선형근사 분석법은 평문과 암호문의 연관관계로부터 키에 대한 정보를 추출하는 기지 평문 공격이다. 그리고 연관키 공격은 키사이의 관계와 해당되는 키들에 대한 기지 평문 또는 선택평문으로 키를 이끌어 내는 방법이다. 이와 같이 현존하는 블록 암호알고리즘에 대한 모든 분석 기술은 평문과 암호문, 키와 암호문, 평문, 키와 암호문 사이에 존재하는 특별한 연관관계를 발견함으로써 개발되었으며, 추후 개발되는 블록 암호알고리즘에 대한 분석 기술도 이 범주를 벗어나지는 않을 것이다.

다. 분석 항목 분류

분 류	소 분 류
키검색 분석	전수검색
	Table Look-Up
	TMTO
키특성 분석	취약키/동치키/보수특성
	연관키
	Slide
차분 분석	기본 분석
	조건부 차분
	부정 차분
	불능 차분
	부메랑
	고계 차분
선형근사 분석	기본근사
	Davies
	다중 선형근사
	선형/비선형근사
	일반선형근사
	Partition / Mod n
기 타	보간 다항식 공격
	차분 · 선형근사분석
부가정보 이용 공격	DPA
	DFA
	Timing 공격
난수 통계특성 검정	아래 표 참조

난수 통계특성 검정법	
중 분 류	소 분 류
주요 통계 검정법	도수 검정
	도수-m 검정
	binary derivative 검정
	m차 binary derivative 검정
	change point 검정
	poker 검정 (Hamming weight 검정)
	runs 검정
	수열복잡도 검정
	선형복잡도 검정
	serial검정
	자기상관 검정
	cuppon collect's 검정
	층돌쌍 검정
엔트로피 검정법	universal검정
Spectral 검정법	spectral검정
기 타	gap 검정
	permutation 검정
	maximum 검정

제 4 장 안전성 분석 항목별 특성 및 분석 조건

가. 키검색 분석

	전수검색	Table Look-up Attack	Time-Memory Trade-Off Cfyptanalysis
조건	하나의 암호문 혹은 한 쌍의 평문 · 암호문	한 쌍의 평문 · 암호문	한 쌍의 평문 · 암호문
사전계산	불필요	필요	필요
공간 복잡도	0(1)	$0(2^{n})$	$0(2^{\frac{2}{3}n})$
시간 복잡도	$0(2^{n})$	0(1)	$0(2^{\frac{2}{3}n})$

1) 전수검색

가) 개념

하나의 암호문 Y_0가 있을 때 모든 가능한 키를 대입하여 Y_0가 평문으로 복호화 되는지를 검사하는 방법과 하나의 평문 X_0와 그에 해당하는 암호문 Y_0가 있을 때 모든 가능한 키를 대입하여 평문 X_0가 암호문 Y_0되는 지 검사하는 방법 두 가지가 있다.

이론적으로 키를 찾을 수 있으나 실제적으로 계산복잡도, 즉, 암호화 및 복호화 과정을 계속적으로 실행하여야 되는 시간상 불가능한 점이 존재하고 주어진 데이터가 암호문이라는 것만을 알고 있을 때 키들을 대입하여 복호화된 출력문이 우리가 원하는 평문인지를 확인 할 방법이 없다라는 단점이 있다.
키 전수검색 분석은 계산능력과 직결되기 때문에 분석을 수행하는 컴퓨터의 계산 능력과 비용대비 키 탐색의 효율성 또한 고려 대상이 된다.

나) 공격방법

(1) 하나의 암호문 Y_0가 있을 때

주어진 하나의 암호문 Y_0와 모든 가능한 키 K_i를 차례로 복호화 알고리즘에 대입하여 출력문이 평문형태로 인식될 수 있을 때까지 검사하는 방법이다.

$While\ i \leq 2^n$

$X_i = D(K_i, Y_0)$

$If\ X_i\ plain\ Text,\ then\ Stop$

(2) 하나의 평문 X_0와 그에 해당하는 암호문 Y_0가 있을 때

주어진 평문 X_0와 모든 가능한 키 K_i를 암호알고리즘에 차례로 대입하여 출력문이 Y_0가 될 때까지 검사하는 방법이다.

While $i \leq 2^n$

$Y_i = E(K_i, X_0)$

If $Y_i = Y_0$, *then Stop*

Else $i \leftarrow i+1$

다) 본 공격법에 대응하는 저항논리

키의 길이를 길게 하는 것이다(현재 75~90 권고!).

라) S/W구현 가능성

구현은 가능하나 키를 찾는 시간제약조건에 따라 가능성이 구분된다.

마) 분석조건

고속 암·복호화 모듈과 부정확한 키를 찾을 경우 발생가능 암호문 혹은 한 쌍의 평문·암호문

2) Table Look-Up 공격

가) 개념

암호문이 주어졌을 때 키 전수검색의 단점은 가능한 모든 키를 암호알고리즘에 대입하여 복호화하여야 하는 시간상의 단점을 극복하기 위하여 사전에 특정평문 X_0과 가능한 키 K_i를 가지고서 암호문들을 발생시켜 저장하였다가 암호문 Y_0가 주어졌을 때 그 암호문에 해당되는 키를 탐색하는 것이다. 그러나 계산량이 없다라는 장점이 있지만, 저장메모리 소요량이 많다는 단점이 존재하며 table look-up 소요시간(탐색시간) 또한 고려하여야 하기 때문에 키 전수조사보다 비용면에서 효과적이지는 못하다.

나) 공격방법

주어진 하나의 평문 X_0에 모든 가능한 키 K_i를 암호알고리즘에 대입시켜 그 키와 키에 의한 출력문 쌍을 Table 형태로 저장하여 암호문이 주어지면 그 암호문에 해당되는 출력문을 사전에 생성한 Table에서 탐색하여 그에 해당되는 키를 찾는 방법으로 다음과 같이 사전계산단계와 탐색단계로 나누어진다.

< 사전계산 >

단계 1. 주어진 하나의 평문 X_0선택

단계 2. 모든 $1 \le i \le 2^n$에 대해서 Table ($Y_i K_i$)를 생성 $Y_i = E(K_i, X_0)$

< 탐색 >

암호문 Y에 대해서 Table에 있는 ($Y_i K_i$)를 탐색하여 암호문 Y에 대한 키를 찾는다.

다) 본 공격법에 대응하는 저항논리

키의 길이를 길게 하는 것이다.

라) S/W구현 가능성

구현은 가능하나 키를 찾기까지의 저장되는 메모리 용량 제약 조건에 따라 가능성이 구분된다.

마) 분석조건

대용량 메모리, 한 쌍의 평문 · 암호문

3) Time Memory Trade-off 공격

가) 개념

1980년 Hellman은 Table Look-Up방법에서의 메모리 사용량을 줄이면서 한번의 암호화로 여러 개의 키를 탐색할 수 있는 Time-memory Trade-Off(TMTO) 공격법을 발표하였다. 이는 계산량을 늘려가면서 키를 탐색하는 방법으로 하나의 기지평문과 여러 개의 look-up table들로써 공격하는데 다음과 같이 사전계산과 탐색 두 가지 단계로 진행된다.

나) 공격방법

o 블록 크기 : m

o 키 길이 : n

o 암 · 복호화 함수 E : $Z_2^m \times Z_2^n \rightarrow Z_2^m$

o 암호화 키(correct key) : $K_0 \in Z_2^n$

o 선택평문(Chosen plaintext) : $X_0 \in Z_2^m$

o 암호문(Ciphertext) : $Y_0 (= E(K_0, X_0)) \in Z_2^m$

o look-up tables의 수 : l

o 조정 함수(Adjustment function) $R^{(s)} : Z_2^m \rightarrow Z_2^n$

o 수정된 함수 $f^{(s)}(K) = R^{(s)}(E(K, X_0)), s = 1, \cdots, l$

o $m \times l$ 키 : $K_{1,0}^{(s)}, K_{1,0}^{(s)}, \cdots K_{m,0}^{(s)}, s = 1, \cdots, l$

o 초기 벡터 $I^{(s)} = \begin{bmatrix} K_{1,0}^{(s)} \\ \vdots \\ K_{m,0}^{(s)} \end{bmatrix}$

o $K_{ij}^{(s)} = f^{(s)}(K_{i,j-1}^{(s)}), i = 1, \cdots, m, j = 1, \cdots, t-1, s = 1, \cdots, l$

$$(i, s) \neq (i', s') \Rightarrow K_{i,0}^{(s)} \neq K_{i,0}^{(s)}, (i = 1, \cdots, m, s = 1, \cdots, l)$$

o 탐색 행렬 $A^s = \begin{bmatrix} K_{1,0}^{(s)} & \cdots & K_{1,t=1}^{(s)} \\ \vdots & & \vdots \\ K_{m,0}^{(s)} & \cdots & K_{m,t=1}^{(s)} \end{bmatrix}$

o 탐색 벡터 $B^{(s)}$

$$B_t^{(s)} = \begin{bmatrix} K_{1,t}^{(s)} \\ \vdots \\ K_{m,t}^{(s)} \end{bmatrix}, \quad K_{i,t}^{(s)} = f^{(s)}(K_{i,t-1}^{(s)}), \quad i = 1, \cdots, m \ , \ s = 1, \cdots, l$$

o Look-up table $D^{(s)} = \begin{bmatrix} K_{\sigma(1),0}^{(s)} & K_{\sigma(1),t}^{(s)} \\ \vdots & \vdots \\ K_{\sigma(m),0}^{(s)} & K_{\sigma(m),t}^{(s)} \end{bmatrix}$

where $\{ \sigma(1), \cdots, \sigma(m) \} = \{1, \cdots, m\}$ such that

$$\sigma(i) \leq \sigma(j) \rightarrow K_{\sigma(i),t}^{(s)} \leq K_{\sigma(j),t}^{(s)}$$

o 수정된 키 $K_r^{(s)}$

$$K_1^{(s)} = R^{(s)}(Y_0) \in Z_2^N$$

$$K_r^{(s)} = f^{(s)}(K_{r-1}^t) \in Z_2^N \quad r = 2, \cdots, t, \quad s = 1, \cdots, l$$

< 사전계산 >

임의의 l을 선택하여 평문 암호문의 크기인 m개와 함께 $m \times l$개의 키들을 $K_{1,0}^{(s)}, K_{1,0}^{(s)}, \cdots K_{m,0}^{(s)}$와 같이 초기에 선택하고 이를 초기 벡터 $I^{(s)}$라 하자. $I^{(s)}$로부터 입출력이 키의 크기와 동일하도록 주어진 암호알고리즘을 변형하여 새로운 키들을 $K_{ij}^{(s)} = f^s(K_{i,j-1}^{(s)}), i = 1, \cdots, m, j = 1, \cdots, t, s = 1, \cdots, l$과 같이 생성하여 벡터 $B_j^{(s)}$, $j = 1, \cdots, t$들을 생성한다. 여기서 $I^{(s)} = B_0^{(s)}$이고 탐색 벡터를 $B_t^{(s)}$ 라 하자. 그리고 생성된 벡터들을 $[B_0^s, \cdots, B_{t-1}^s]$와 같이 차례로 배열한 행렬을 탐색 행렬 $A^{(s)}$라 하

자. 이 때 저장되는 데이터는 탐색 행렬의 첫 번째 열 $I^{(s)}=B_0^{(s)}$과 탐색 벡터 $B_t^{(s)}$만 저장하는 look-up table $D^{(s)}=[B_0^{(s)},B_t^{(s)}]$을 구성한다.

상기와 같은 방법을 알고리즘화 하면 다음과 같다.

단계 1. $m\times l$ Keys를 선택한다 $K_{1,0}^{(s)},K_{1,0}^{(s)},\cdots K_{m,0}^{(s)}$

단계 2. 선택된 $I^{(s)}$로부터 $K_{ij}^{(s)}=f^{(s)}(K_{i,j-1}^{(s)})$를 계산한다.

$$i=1,\cdots,m,j=1,\cdots,t-1,s=1,\cdots,l$$

단계 3. 탐색 벡터 $B_j^{(s)},j=1,\cdots,t$를 계산한다.

(탐색 행렬 $A^{(s)}=[B_0^s,\cdots,B_{t-1}^s]$ 이다.)

단계 4. Look-up table $D^{(s)}=[B_0^{(s)},B_t^{(s)}]$만 저장한다.

< 탐색 >

주어진 암호문 Y_0와 그에 해당하는 평문 X_0에서 사전계산에서 저장되어 2개의 열로 구성된 look-up table $D^{(s)}=[B_0^{(s)}, B_t^{(s)}]$의 원소들 중에서 암호화 키 K_0를 탐색하는 단계로써 다음과 같이 진행한다.

우선 주어진 암호문 Y_0를 입력으로 하는 수정된 키 $K_1^{(s)}= R^{(s)}(Y_0)$ 를 계산한다. 계산된 수정된 키 $K_1^{(s)}$ 가 저장된 Look-up table $D^{(s)}$의 두 번째 열 벡터인 $B_t^{(s)}$에 존재하는 지를 탐색한다. 만약 없으면 새로운 수정된 키 $K_2^{(s)}= f^{(s)}(K_1^{(s)})$를 계산하여 $K_2^{(s)}$가 저장된 Look-up table $D^{(s)}$의 두 번째 열 벡터인 $B_t^{(s)}$에 존재하는 지를 다시 탐색한다. 이러한 과정을 연속하여 반복하는 과정에서 수정된 키 $K_p^{(s)}= f^{(s)}(K_{P-1}^{(s)})$가 Look-up table $D^{(s)}$의 두 번째 열 벡터인 $B_t^{(s)}$존재하면, 즉, $B_t^{(s)}$의 어떤 i번째 원소에 대해서 $K_p^{(s)}= K_{i,t}^{(s)}$이면, $K_0= K_{i,t-1}^{(s)}$이거나 false alarm경우인 $K_{i,t}^{(s)}$ 역상 $K_{i,t}^{(s)}= f^{(s)}(K^*) = f^{(s)}(K^{**})$이 두 개 이상인 경우가 된다. Look-up table $D^{(s)}$의 첫 번째 열 벡터인 $I^{(s)}= B_0^{(s)}$의 원소 $K_{i,0}^{(s)}$로 부터 $K_{i,t-1}^{(s)}$를 계산하여 $K_{i,t-1}^{(s)}$가 $K_{i,t-1}^{(s)}= K_0$인지를 검사하여 암호화 키 K_0를 찾는 방법이다.

단계 1. 수정된 키 $K_1^{(s)}= R^{(s)}(Y_0)$를 계산한다.

단계 2. $K_1^{(s)}$이 Look-up table $D^{(s)}$의 두 번째 열에 있는가를 검사한다.

$K_1^{(s)}$이 Look-up table $D^{(s)}$의 두 번째 열에 없으면, K_0는 탐색 행렬 $A^{(s)}$의 $t-1$ 열에 없다.

단계 3으로

$K_1^{(s)} = K_{i,t}^{(s)}$이면 다음 중 하나가 만족된다.

(i) $K_0 = K_{i,t-1}^{(s)}$

(ii) $K_{i,t}^{(s)}$ 역상 $K_{i,t}^{(s)} = f^{(s)}(K^*) = f^{(s)}(K^{**})$은 두 개 이상이고 $K_0 \neq K_{i,t-1}^{(s)}$ 그러므로 $K_{i,0}^{(s)}$ 로부터 $K_{i,t-1}^{(s)}$를 계산하여 $K_{i,t-1}^{(s)}$ 가 $K_{i,t-1}^{(s)} = K_0$ 인지를 검사한다.

단계 3. K_0 가 탐색 행렬 $A^{(s)}$의 $t-1$번째 열에 없으면 $K_2^{(s)} = f^{(s)}(K_1^{(s)})$를 계산하여 $K_2^{(s)}$ 가 $B^{(s)}$에 있는 가를 검사한다.

$K_2^{(s)} = f^{(s)}(K_1^{(s)})$이면 $K_{i,t-1}^{(s)} = K_0$ 를 검사한다.

단계 4. K_0가 탐색 행렬 $A^{(s)}$의 $t-2$번째 열에 없으면 $K_r^{(s)} = f^{(s)}(K_{r-1}^{(s)}), r = 3, \cdots, t$를 계산하여 단계 3과 동일하게 검사한다.

다) 본 공격법에 대응하는 저항논리

키의 크기를 길게 하는 것이다.

라) S/W구현 가능성

구현은 가능하나 키를 찾기까지의 저장되는 메모리용량 제약조건과 시간제약조건을 동시에 고려하여야 한다.

마) 분석조건

한 쌍의 평문 · 암호문

나. 키특성 분석

1) 취약키

주어진 키로 두 번 암호화하면 모든 평문이 복원되는 키를 DES형 취약키(weak key)라 한다(즉, $E_K(E_K(P))=P$). 이를 확장하여 두 개의 K_1, K_2가 있을 때 K_1로 암호화하고, 다시 K_2로 암호화하면 평문이 복원되는 키를 준 취약키(semi-weak key)라 한다(즉, $E_{K_2}(E_{K_1}(P))=P$).

블록 암호알고리즘을 해쉬 함수로 사용하는 경우가 있는데, 평문을 키 스케줄에 입력시키고 고정된 상수를 반복적으로 암호화하여 해쉬 값을 구하는 형태가 있다. 이러한 방식에서 취약키가 있으면 충돌 쌍을 찾는데 사용될 수 있다. 그러므로 취약키와 준 취약키는 없는 것이 바람직하다.

한편 각종 공격 방식이 적용되었을 때 쉽게 해독되는 키도 취약키라 할 수 있는데, 이러한 키는 IDEA와 같이 산술 연산이 주로 사용되는 경우에 종종 발생하며, 이러한 키들은 대응되는 공격 방식에 따라 면밀히 검토해야 할 대상으로 평가자에 의해 조사되어야 한다.

2) 동치키

주어진 키 K_1과 K_2가 동일한 평문에 대해 동일한 암호문을 출력할 때, K_1과 K_2를 동치키(equivalent key)라 한다(즉, $E_{K_1}(E_{K_2}(P))=P$). 동치키가 존재하면 키 검색 공간은 그만큼 줄어든다. 따라서, 키 길이를 정확하게 결정하려면 동치키를 동일한 키로 간주하여 계산하여야한다. 동치키와는 다르지만 보수 특성과 같이 몇 개의 평문, 키, 암호문에 연관성이 있으면 키 검색 공간을 줄일 수 있는데 키 검색 공간 측면에서 키 길이를 조사하여야 한다.

3) 보수 특성

평문 P의 보수(bitwise complement)를 키 K의 보수로 암호화했을 때 암호문 C의 보수가 출력되면, 그 암호알고리즘은 보수 특성(complementation property)을 가진다고 말한다(즉, $E_K(P)=C$일 때, $E_{\overline{K}}(\overline{P})=\overline{C}$). 주어진 블록 암호알고리즘이 보수 특성을 가지면 키 전수 검색시 검색 공간을 반으로 줄일 수 있기 때문에 피해야 하는 특성이다.

4) 연관키 공격

연관키 공격(related key)이란 서로 다른 두 키 사이의 연관 관계를 알고 있지만 키 자체를 모를 때 각 키로부터 발생된 평문, 암호문 쌍을 가지고 키를 알아내는 공격 방식이다. 연관키 공격은 상당히 강력한 공격이지만 연관 관계에 있는 각각의 키로 암호화된 평문, 암호문 쌍을 구한다는 가정이 매우 비현실적이기 때문에 실질적인 공격 방법으로 보기는 어렵다. 다만 키 사용이 K, K+1 K+2, · · · 와 같이 규칙적으로 변하는 경우나 키 공유 프로토콜에 개입하여 인위적으로 키를 조작할 수 있는 경우에 적용이 될 수 있다. 그러나 암호알고리즘이 아주 특별한 환경에서 사용되거나, 해쉬 함수로 사용될 때에는 연관키 공격이 적용될 수도 있기 때문에 평가 측면에서 고려할 필요는 있다. 연관키 공격 관

점에서 제시되는 키 스케줄 설계 원칙은 다음과 같다.
o 라운드 키로부터 키를 복원할 수 없어야 한다.
o 키의 선형 변환으로 라운드 키가 발생되어서는 안 된다.
o 키의 변화로부터 라운드 키의 변화를 제어할 수 없도록 하여야 한다.
o 라운드 키 발생이 키의 단순 쉬프트로 발생되는 경우는 LOKI와 같이 연관키 공격에 취약할 수 있으므로, 이를 방지하기 위한 조치로 각 키의 모든 비트는 거의 모든 라운드 키 발생에 참여하여야 하며, 똑같은 방법으로 라운드 키가 발생되는 것은 피해야 한다.
가장 단순한 연관키 공격으로 암호알고리즘을 Black box로 고려하는 방식이 있는데, 암호알고리즘이 k비트의 키를 사용한다면 2n개의 선택키와 하나의 선택 평문에 대해 2k-n의 암호화로 키를 찾을 수 있음이 알려져 있다. 여기서 연관키의 선택에 따른 어려움 정도를 가정하여 만약 연관키 선택이 쉽다면 Black Box 공격에 강할 정도로 키 길이가 길어야 한다는 것이 평가 항목에 포함되어야할 것이다.
한편 연관키 공격 관점에서 키 스케줄이 없이 모든 라운드 키가 독립적인 키인 경우는 피해야 한다. 이는 맨 마지막 라운드 키만 다르고 그 외의 라운드는 모두 같은 라운드 키로 이루어진 라운드 키를 선택할 수 있다면 1라운드 공격과 같은 개념이 되어 쉽게 해독되는 특성이 있기 때문이다. 물론 이 조건이 절대적인 평가 항목이 되는 것은 문제가 있고, 보조적인 요소로 고려하여야 한다.

5) Slide 공격

블록 암호알고리즘에서 매 라운드에 입력되는 라운드 키가 동일할 경우 공격하는 기법으로 블록의 크기에만 의존하고 라운드 수에는 무관한 강력한 공격 기법이다. 그러나 모든 라운드 키가 동일하게 설계되는 블록 암호알고리즘은 거의 없으므로 현실적으로 적용될 수 있는 경우는 매우 제한적이다. 다만 운영상의 편리를 위해 라운드 키를 일치시켜 사용하는 경우의 위험성을 지적한다고 볼 수 있다.

<공격 방법>

① 블록의 크기가 n일 때 $2^{2/n}$개의 기지 평문을 얻는다.

② 평문 P_1를 1라운드 암호화시킨 값에 대응되는 평문 P_2를 찾는다.

③ 대응되는 암호문 C_1을 1 라운드 암호화시키면 암호문 C_2가 생성되기 때문에 이러한 특성을 이용하여 키를 찾는다.

이 공격법은 라운드 키가 동일한지 여부를 조사함으로써 적용가능성을 쉽게 판단할 수 있으며, 별도의 소프트웨어 개발은 필요가 없다.

다. 차분 분석

1) 기본 차분 분석

가) 개요

평문이 한 비트 변하면 출력의 각 비트가 0.5의 확률로 변해야한다는 개념이 SAC(Strict Avalanche Criteria)이다. 이 개념이발전하면 입력이 어떠한 형태로 변하더라도 출력의 각 비트는 확률적으로 반정도 변해야 한다는 PC(Propagation Criteria) 개념이 도출되었다. 그런데 이러한 개념을 암호 분석에 사용한 것이 차분 분석법(Differential Cryptanalysis, DC)이다.

차분 분석법은 Crypto'90에서 발표되었지만 DES를 설계할 당시에 이미 이 공격법을 고려하여 설계하였음이 알려졌고, 1990년이후 대부분의 블록 암호알고리즘이 이 공격에 의해 문제점이 발견되었으며, 최근에 제안되는 블록 암호알고리즘은 선형근사 분석법과 더불어 이 공격에 대한 안전성을 기본적으로 고려하고 있다.

나) 개념

차분 분석법은 평문이 특정한 형태로 변할 때 높은 확률로 암호문이 특정 형태로 변하는 특성이 있으면 그 특성을 이용하여 주어진 암호를 해독하는 방법으로 선택 평문 공격이라 할 수 있다. 이 때 평문의 변화와 암호문 변화를 차분 특성이라 정의한다.

[차분 특성]

임의의 함수 F의 두 개 입력을 P_1, P_2라 하고 대응되는 출력을 $C_1 C_2$라 하자. 만약 연산 $\odot$에 대해 $P' = P_1 \odot P_2^{-1}$되는 임의의 두 개 입력에 대응되는 출력이 확률 p로 $C' = C_1 \odot C_2^{-1}$가 되면 (P', C', p)를 함수 F에 대한 차분 특성이라 한다.

보통 입력과 키가 결합되는 연산에 대한 차분 특성을 사용하는데 이는 차분 특성이 키에 독립적인 값이 되게 함으로써 키 요소를 고려하지 않아도 되도록 하는 것이다. Biham과 Shamir는 주어진 블록 암호알고리즘의 라운드 함수에 대한 차분 특성을 구하고 이를 결합하여 n 라운드 차분 특성을 구하여 블록 암호알고리즘을 분석하였다.

주어진 블록 암호알고리즘을 공격하기 위해서는 차분 특성 (P', C', p)에 대응되는 right pair의 개념이 필요하다. 두 개의 평문 P_1, P_2가 $P' = P_1 \odot P_2^{-1}$를 만족하고 대응되는 출력이 $C' = C_1 \odot C_2^{-1}$를 만족할 때 이러한 암호문 쌍을 right pair라 한다. 만약 이러한 조건을 만족하지 않으면 wrong pair라 한다.

대부분의 블록 암호알고리즘이 키가 결합될 때 XOR 연산을 사용하기 때문에 통상적으로 차분 특성을 고려할 때는 $P' = P_1 \oplus P_2^{-1}$에 해당하는 특성을 조사한다.

다) 공격 방법

만약 주어진 n 라운드 차분 특성의 right pair를 알 수 있고, n+1 라운드 후의 암호문이 주어진다면 n+1 라운드에서 라운드함수의 출력 변화를 알 수 있다. 그러므로 n+1 라운드의 키를 추출할 수 있게 된다. 차분 분석법은 right pair를 확률적으로 알 수 있으며, right pair를 효과적으로 결정할 수 있을 경우 쉽게 공격할 수 있게 된다.

차분 분석법은 확률이 큰 차분 특성 (P', C', p) 을 이용하여 다음과 같이 키를 찾아낸다.

[차분 분석법]

단계 1. 두개의 평문 변화가 P'인 많은 수의 평문 쌍에 해당하는 암호문 쌍을 획득한다.

단계 2. 주어진 암호문 쌍이 wrong pair로 확정되면 버린다.

단계 3. 구하고자 하는 키의 비트 수에 해당하는 메모리를 초기화시킨다.

단계 4. 확률적으로 구해진 right pair로부터 가능한 키를 추출하여 해당키에 해당하는 메모리의 값을 1 증가시킨다.

단계 5. 메모리 값 중 가장 큰 것에 해당하는 키를 올바른 키로 판정한다.

주어진 차분 특성을 이용하여 키를 복원할 때 right pair를 정확히 알 수 없기 때문에 단계 2.에서 버려지지 않은 wrong pair는 틀린 키를 추출하게 된다. 따라서 위의 공격법이 성공하기 위해서는 올바른 키의 값이 증가하는 틀린 키의 값이 증가하는 속도에 비해 커야 한다. 이러한 관점에서 평균적으로 증가하는 키의 개수에 대한 right pair 개수의 비율을 Signal-to-noise ratio라 하고, S/N으로 표기한다. S/N이 1 보다 커야 차분 분석법은 성공한다.

n 라운드 차분 특성을 이용하여 n+1 라운드의 키를 찾으면 1R-attack이라 하고, n+k 라운드의 키를 찾으면 kR-attack이라 한다. DES의 경우 n+3 라운드까지 공격할 수 있어 3R-attack이 가능하다. 그러므로 16라운드 DES를 공격하기 위해서는 13라운드의 차분 특성이 있어야 한다.

라) 적용 환경

차분 분석법을 적용하기 위해서는 다음의 조건을 만족해야 한다.

- 확률이 높은 차분 특성이 존재해야 한다.
- 효과적으로 wrong pair를 버릴 수 있는 방법이 있어야 한다.
- S/N가 1 보다 커야 한다.

마) 소프트웨어 개발 가능성

DES에서는 최적의 차분 특성을 구하는 방법이 제안되어 있으나, 임의의 블록 암호알고리즘에 대한 최적의 차분 특성을 구하는 소프트웨어의 개발은 거의 불가능 해 보인다. 주어진 블록 암호알고리즘에 대한 최적의 차분 특성은 보통의 경우 구하기 어려우며, 구할 수 있다고 하여도 각 알고리즘 별로 별도의 소프트웨어가 개발되어야 할 것이다.

2) 조건부 차분 분석

가) 개요

차분 분석법이 알려진 후 이 공격법에 안전한 블록 암호알고리즘을 설계할 목적으로 키에 의해 데이터의 교환이 이루어지는 논리가 제안되었는데, 이러한 형태의 블록 암호알고리즘에 적용 가능한 공격 기법이다. 기본 개념은 키의 공간을 축소하여 데이터의 교환을 예측함으로써 차분 분석법을 적용하는 것(Conditional DC)으로 Lucifer, RDES, ICE등이 적용되었다.

나) 개념

키에 의해 가변되는 논리에서 가변성을 없애기 위해 키의 특정 비트를 예측하고 예측된 키에 의해 고정된 논리에 대하여 차분 분석법을 시행하는 방법으로 관련된 몇 가지 개념은 다음과 같다.

[키에 대한 차분 특성]

고정된 키 K에 대한 차분 특성 $Q_K=(P', C', p)$는 평문 변화가 P'인 임의의 입력 쌍이 고정된 키 K에 대하여 출력의 변화가 C'로 될 확률 p로 정의한다.

[키 집합에 대한 차분 특성]

키 집합 U에 대한 차분 특성 Ω_u의 확률을 Ω에 있는 키 K에 대한 차분 특성 Ω_K의 확률 중 최소값으로 정의한다.

[조건부 차분 특성]

조건부 차분 특성을 (Ω_u, U, p)로 정의하는데 여기서 Ω_u는 차분 특성, U는 키 공간의 부분집합, p는 U에 대한 Ω_u의 확률을 의미한다.

[키 비율]

조건부 차분 특성 (Ω_U, u, p)의 키 비율이란 전체 키 공간에 대한 U의 비율, 즉 $\frac{|U|}{|K|}$를 의미한다.

U의 크기가 작아지면 조건부 차분 특성의 확률은 커지지만 키 비율은 작아져 조건부 차분 특성을 이용한 공격에서 찾을 수 있는 키 공간이 작아지고, U의 크기를 증가시키면 조건부 차분 특성을 이용한 공격이 어려워진다.

다) 공격 방법

기본적인 공격 방법은 차분 분석법과 동일하다. 다만 키에 의해 가변되는 논리를 예측할 수 있으면서 키 공간을 최대한 크게 하는 조건부 차분 특성을 구하는 것이 관건이다.

라) 적용 환경

개념적으로는 키에 의해 가변되는 모든 논리에 적용이 가능하지만 현실적인 공격이 가능하기 위해서는 키 비율이 큰 조건부 차분 특성을 구할 수 있는 경우이다. 즉, 모듈러 곱셈과 같이 키 각각에 대한 논리가 모두 다른 경우에는 적용하기가 어려우며, 키의 특정 비트에 의해 데이터가 교환되는 논리와 같이 키의 특정 비트만 고정하면 동작 방식을 예측할 수 있는 경우에 적용될 수 있다.

3) 부정 차분 분석

가) 개요

기존의 차분 분석법은 특정한 라운드 후의 출력 변화를 모두 예측하여 공격하는데 실제 공격할 때에는 출력 변화를 모두 알 필요는 없으며 특별한 경우에는 한 비트만 알아도 공격할 수 있다. 이러한 관점에서 출력 변화의 특별한 비트만을 예측하는 개념으로 부정 차분(Truncated Differential)이 정의되었다.

부정 차분 특성을 이용하면 키가 작용하는 연산과 차분 특성의 연산이 달라도 쉽게 적용할 수 있어 IDEA나 SAFER와 같은 블록 암호알고리즘의 분석에 적용되었다.

나) 개념

차분 분석법은 다음의 3단계로 공격이 이루어진다.

1. 확률이 높은 차분 특성을 구한다.
2. 차분 특성의 입력 변화에 대응되는 선택 평문을 구하고, 이 선택 평문에 대응되는 암호문 쌍 중 wrong pair를 버린다.
3. wrong pair로 버려지지 않은 암호문 쌍으로부터 마지막 라운드 함수의 출력 변화를 예측하고 이로부터 가능한 키를 추출한다.

따라서 차분 분석법이 성공하기 위해서는 확률이 높은 차분 특성이 있어야 하며, wrong pair를 효과적으로 버릴 수 있는 기법이 필요하다. 즉, 차분 특성의 확률을 크게 할 목적으로 부정 차분 특성의 개념이 도입되었다.

[정의(부정 차분 특성)]

i 라운드 차분 특성을 $\Omega=(\Omega_o, \Omega_T, p)$라 할 때, $\bar{\Omega}_p$가 Ω_p의 특정 비트열이고 $\bar{\Omega}_T$가 Ω_T의 특정 비트열이면 $\bar{\Omega}=(\bar{\Omega}_p, \bar{\Omega}_p)$를 i 라운드 부정 차분 특성이라 한다.

부정 차분 특성은 개념적으로는 예측되지 않는 비트에 대한 모든 가능한 차분 특성의 합이라 할 수 있기 때문에 확률이 그만큼 클 수 있다. 그러나 예측되지 않는 비트가 증가하기 때문에 기존의 차분 분석법에서 wrong pair를 버릴 수 있는 수단이 적어진다. 따라서 기존의 차분 분석법에서 wrong pair 여과 단계의 효율성을 희생한 것에 비하여 차분 특성의 확률이 높아짐으로써 얻을 수 있는 효과가 클 때 적용 가능하다.

한편 입력의 변화에서도 모든 비트를 지정할 필요가 없기 때문에 특정한 형태의 평문 데이터의 집합인 structure를 이용하여 적은 수의 선택 평문으로도 많은 수의 평문 쌍을 획득할 수 있는 기법을 적용할 수 있는 장점이 있다. 이는 선택 평문의 개수를 줄이는데 효과적으로 사용된다.

다) 공격 방법

기본적인 공격 방법은 차분 분석법과 동일하다. 다만 Structure를 이용하여 필요한 선택 평문을 줄이고 대신에 wrong pair를 여과시키는 단계의 효율성은 감소한다.

라) 적용 환경

IDEA나 SAFER와 같이 키에 작용하는 연산과 차분 특성에서 사용하는 연산이 다른 경우에 효과적으로 적용되었으나 개념적으로는 모든 블록 암호알고리즘에 적용 가능하다.

4) 불능 차분 분석

가) 개요

기존의 차분 분석법에서 wrong pair를 버리는 기법을 극대화시킨 기법이라 할 수 있으며, 최근에 공개된 SKIPJACK 분석과정에서 개발된 공격법이다. 간략한 개념은 절대로 발생되지 않는 차분 특성(불능 차분 특성,Impossible DC)을 찾고 마지막라운드의 키를 가정하였을 때 불능 차분 특성이 발견되면 그 키를 틀린 키로 판단함으로써 올바른 키를 추출하는 방법이다.

이 방법으로 4라운드 IDEA가 분석되었으며 8비트에서 32비트로 가는 S-box를 이용하는 블록 암호알고리즘에 대한 취약점이 제기되었다. 대표적인 예로 Khufu와 Khafre에 대한 암호 분석이 시도되었다.

나) 개념

차분 분석법은 상대적으로 높은 확률을 갖는 차분 특성을 사용하여 키를 찾아내는 방법이다. 공격 과정은 주어진 암호문 쌍에서 차분 특성을 만족하는 암호문 쌍을 선택하고(wrong pair의 여과), 이로부터 구성된 마지막 라운드의 입출력 차분 특성을 주는 키를 세면서 가장 많이 세어진 키를 올바른 키로 선택한다.

그런데 wrong pair의 여과 과정을 살펴보면 주어진 차분 특성이 만족해야 하는 조건을 조사하는 것과 마지막 라운드의 차분 특성이 발생하지 않는 경우 이를 버리는 과정으로 세분할 수 있다.

라운드 함수의 차분 특성이 절대로 발생하지 않는 경우를 확장하여 임의의 라운드에 대한 차분 특성에서 확률이 '0'인 차분 특성을 불능 차분 특성이라 하며 이러한 차분 특성을 이용하여 공격하는 기법을 불능 차분 분석법이라 한다.

다) 공격 방법

개념적인 공격 방법은 다음과 같다.

1. 불능 차분 특성을 찾는다.
2. 불능 차분 특성의 입력 차분을 만족하는 선택 평문 쌍에 대한 암호문 쌍을 얻는다.
3. 마지막 라운드에 대한 키를 가정하고 주어진 암호문 쌍을 복호화하여 불능 차분 특성을 만족하는지 검사한다.
4. 불능 차분 특성을 만족하는 라운드 키는 버리고, 이 과정을 반복하여 최종적으로 남는 키를 올바른 키로 택한다.

라) 적용 환경

불능 차분 특성이 존재하면 항상 적용할 수 있으며, 공격의 핵심은 가장 긴 라운드의 불능 차분 특성을 찾는 것과 이를 이용하여 공격할 수 있는 라운드 수를 증가시키는 것이다.

CAST의 S-box와 같이 입력 비트 보다 출력 비트가 많은 암호 논리가 사용되었을 때에는 차분 특성에서 확률 0이 되는 경우가 많으므로 불능 차분 분석법 관점의 검토가 필요하다.

5) 고계 차분 분석

차분 특성에 대한 차분 특성을 고려할 수 있는데 이러한 특성을 고계 차분 특성(Higher Order DC)이라 한다. 차분 특성을 부울 함수로 고려하면 원래의 함수보다 대수적인 차수가 작아지는데 대수적인 차수가 작은 함수로 설계된 블록 암호알고리즘은 고계 차분 특성을 이용한 공격에 취약하다. 즉, 블록 암호알고리즘에서 라운드 수를 반복하여도 대수적 차수가 크게 증가하지 않으면 주어진 암호의 마지막 라운드를 복호화하여 고계 차분을 구해서 상수가 되는 지 확인함으로써 라운드 키를 구할 수 있다.

이 공격은 라운드 함수의 대수적 차수에 의존하기 때문에 안전성평가를 쉽게 할 수 있을 것이다.

6) 부메랑 공격

가) 개요

불능 차분 공격은 중간에 일치하지 않는 차분 특성, 즉 불능 차분 특성을 이용하는 방법인데 부메랑 공격은 중간 라운드까지의 차분 특성을 찾고 그에 대응되는 암호문 쌍을 선택하여 중간라운드의 차분 특성이 일치하는지 조사함으로써 이루어진다. 따라서 선택 평문/능동 선택 암호문 공격이라 할 수 있다.

나) 개념

부메랑 공격은 주어진 암호의 중간 라운드에서 일치하는 quartet 구조를 사용한다. 주어진 암호를 반으로 나누었을 때, 앞부분을 E_0라하고, 뒷부분을 E_1이라 할 때 E_0의 차분 특성이 $\Delta P \rightarrow \Delta T$이고 E_1^{-1}의 차분 특성이 $\Delta C \rightarrow \Delta T$인 다음과 같은 차분 특성을 이용하여 공격한다.

① $P_1 \oplus P_2 = \Delta P, E_0(P_1) \oplus E_0(P_2) = \Delta T$인평문 P_1, P_2를 선택

② 암호문 $C_1\ C_2$에 $C_1 \oplus C_3 = \Delta C, C_2 \oplus C_4 = \Delta C$이 되는 암호문 $C_3\ C_4$를 선택

③ 그러면 $E_1^{-1}(C_1) \oplus E_1^{-1}(C_3) = \Delta T, E_1^{-1}(C_2) \oplus E_1^{-1}(C_4) = \Delta T$이 성립

④ 다음이 성립하므로 E_0에 대한 차분 공격 가능

$E_0(P_3) \oplus E_0(P_4) =$

$= E_0(P_1) \oplus E_0(P_2) \oplus E_0(P_1) \oplus E_0(P_3) \oplus E_0(P_2) \oplus E_0(P_4)$

$= E_1^{-1}(C_1) \oplus E_1^{-1}(C_2) \oplus E_1^{-1}(C_1) \oplus E_1^{-1}(C_3) \oplus E_1^{-1}(C_2) \oplus E_1^{-1}(C_4)$

$= \Delta T \oplus \Delta T \oplus \Delta T = \Delta T$

※ 부메랑 분석법은 1999년도에 발표되어 지속적인 연구 검토가 필요하다.

라. 선형근사 분석

1) 기본 선형근사 분석

가) 개념

선형근사 분석법(Linear Cryptanalysis)은 임의의 주어진 평문P와 키 K 및 이에 대응하는 암호문 C에 대하여 확률 p= 1/2 + a로 다음과 같은 선형 근사식이 존재할 때 적용 가능하다.

$$P \cdot \Lambda_p \oplus C \cdot \Lambda_c = K \cdot \Lambda_K$$

위의 식이 높은 확률로 성립하면, maximum likelihood법을 이용한 아래의 알고리즘으로 키의 1 비트를 의미 있게 추정할 수 있다.

[개념적인 선형근사 분석 방법]

단계 1. 주어진 모든 기지 평문과 이에 대응되는 암호문 쌍으로부터 선형근사식의 좌변을 계산한다.

단계 2. 이 값이 0이 된 평문수가 전체 평문의 반 이상이 되면 $K \cdot \Lambda_K = 0$ (p > 1/2 일 때), 또는 1(p < 1/2 일 때)로 추정하며, 값이 0이 된 평문수가 전체 평문의 반 이하 되면 $K \cdot \Lambda_K = 1$ (p > 1/2 일 때), 또는 0 (p < 1/2 일 때)으로 추정한다.

이 추정이 성공할 확률은 기지 평문 수 N과 선형근사식의 성립 확률 p로 결정되며 N 또는 |p-1/2| 이 클수록 공격에 성공할 가능성이 커진다. 선형 근사식 중 |p-1/2| 가 최대가 되는 것을 최량 표현이라 부르며, 그 성공 확률을 최량 확률이라 부른다.

n 라운드의 암호를 공격하기 위해서는 n-2 라운드의 최량 표현을 사용한다. 즉, 1, n 라운드는 첫 번째와 n번째 라운드 함수 F에 입력되는 라운드 키 K_1, K_n만을 이용한다는 점을 이용하여 근사식 중에 F함수를 넣어 계산한다. 그 결과는 다음 식과 같으며 우리는 이 식을 n-2 라운드의 최량 확률로 성립시키도록 한다.

$$P \cdot \Lambda_p \oplus C \cdot \Lambda_c \oplus F_1(P, K_1) \cdot \Lambda_1 \oplus F_n(C, K_n) \cdot \Lambda_n = K \cdot \Lambda_K$$

그런데 만일 이 식에 틀린 키가 대입되면 거의 성립하지 않을 것이다. 즉, 이 식의 성립 확률이 거의 1/2이 될 것이다. 따라서 다음의 알고리즘에서와 같이 maximum likelihood 법을 이용하여 키를 의미있게 추정할 수 있다.

[n-2 라운드 선형 근사식을 이용한 선형근사 분석 방법]

단계 1. K_1, K_n의 각 후보값에 대한 카운터를 설정하여 '0'으로 초기화한다.

단계 2. 주어진 각 기지 평문과 이에 대응하는 암호문 쌍에 대하여 다음을 실행한다.

- K_1, K_n의 각 후보값에 대하여 선형 근사식의 좌변을 계산하고, 그 결과가 '0'으로 된 키에 해당하는 카운터 값에 1을 증가시킨다.

단계 3. 모든 카운터의 값에서 최대치 T_{max}와 최소치 T_{min}을 비교하여 기지 평문 수 N에 대하여

- $|T_{max} - N/2| > |T_{min} - N/2|$ 이면, T_{max}에 대응하는 키를 선택한다.

- | T_{min} - N/2 | > | T_{max} - N/2 | 이면, T_{min}에 대응하는 키를 선택한다.

선형근사 분석법을 적용하기 위해서는 확률이 높은 선형근사식이 존재해야 한다. 따라서 선형근사 분석법에 대한 안전성을 평가하기 위해서는 확률이 큰 선형근사식을 찾아야 하는데 일반적인 방법으로 주어진 선형근사식을 구하는 것은 어려울 것으로 예측되며, DES와 같이 특수한 형태일 경우에 선형근사식을 계산할 수 있는 소프트웨어를 개발할 여지는 존재한다.

2) Davies 공격

DES와 같이 라운드 함수에서 입력이 확장되고 키가 XOR되는 경우에는 동일한 입력 비트에 다른 키 비트가 XOR된다. 이러한 경우에 동일한 입력 비트를 이용한 두개의 선형근사식을 XOR하면 키 비트와 암호문과의 선형근사식이 존재하게 된다. 그러면 암호문만을 가지고 키 비트를 추정할 수 있는데 이러한 공격 방법을 Davies 공격이라 말한다.

Davies 공격은 DES와 같이 인접한 두개의 S-box에 동일한 입력이 작용하지만 키 비트는 다른 경우에 적용될 수 있는 공격 기법이다. 이러한 개념을 발전시켜 라운드 함수가 전사가 아닌 경우에도 라운드 함수의 출력에 대한 통계적인 특성을 이용하여 공격할 수 있기 때문에 라운드 함수가 전사 함수가 아닌 경우 이 공격 관점에서 별도의 평가 과정이 있어야 한다.

3) 다중 선형근사 분석

다중 선형근사 분석법(Multiple linear approximation)은 키에 대한 마스크 값은 동일하고 평문과 암호문에 대한 마스크 값은 다른 n개의 선형근사식이 존재할 때, 필요한 기지 평문의 개수를 줄이는 기법으로 제안되었다. 즉, n개의 선형 근사식은 각각의 성립 확률과 분산값을 가지고 있는 통계치 T_i를 정의할 수 있는데 기존의 선형근사 분석법은 각각의 통계치를 이용하여 공격한다. 그런데 이 n개의 선형 근사식 모두를 동시에 사용하는 분석법을 다중 선형근사 분석법이라 말한다.

우선 n개의 선형 근사식 중 i번째의 근사식을 다음과 같이 정의하고 성립 확률을 1/2 + e_i라 하자.

$$P \cdot \Lambda_P^i \oplus C \cdot \Lambda_C^i = K \cdot \Lambda_K$$

분석을 쉽게 하기 위해서 각각의 bias의 값 e_i를 '0'보다 크다고 가정한다(만약 원래의 bias의 값 e_i이 '0'보다 작다면 왼쪽이나 오른쪽 중 하나에 '1'을 XOR하면 바뀐 선형근사식의 bias값은 '0'보다 크다).

[다중 선형근사 분석 알고리즘]

<입력> 획득한 기지 평문의 개수를 N이라 하자.

단계 1. 모든 i에 대해 해당하는 선형근사식의 좌변을 계산하여 '0'이 되는 쌍의 수를 T_i라 한다,

단계 2. 각각의 T_i에 가중치 a_i를 적용하여 다음을 계산한다. 이 때 a_i들의 합은 '1'이다.

$$U = a_1 T_1 + a_2 T_2 + \cdots + a_n T_n$$

단계 3. 만약 U > N/2이면 $K \cdot \Lambda_K$=0이라고 추측하고, 만약 U<N/2이면 $K \cdot \Lambda_K$ =1이라고 추측한다.

이 알고리즘에서 가중치 a_i는 다음과 같이 설정한다.

$$a_i = e_i / (\sum e_i)$$

다중 선형근사 분석법은 선형근사 분석법에서 동일한 키 비트들에 대한 다른 선형근사식이 존재할 때 기지 평문 수를 줄이는 방법이며, 평가측면에서 고려한다면 선형근사 분석법의 보조적인 요소로 고려하면 될 것이다.

4) 선형/비선형 근사 분석

선형근사 분석법은 블록 암호알고리즘에서 각 라운드를 선형근사시켜 전체 암호알고리즘을 선형근사식으로 고려하여 공격한다. 그런데 이러한 공격을 효과적으로 수행하기 위해서는 확률이 큰 선형근사식이 필요하다.

만약 선형근사식 대신에 비선형근사식을 사용한다면 보다 확률이 높은 근사식을 만들 수 있다. 그런데 중간 라운드에서는 XOR의 특성 때문에 비선형근사식을 이용할 수 없다. 단지 첫 번째 라운드와 최종 라운드에서만 비선형근사식을 이용할 수 있는데 이러한 방법을 선 형/비선형 근사 분석법(Linear/Non-linear Approximation)이라 한다.

비선형근사식을 이용할 수 있는 라운드가 제한적이기 때문에 안전성 평가시에는 선형근사 분석법의 보조적인 수단으로 사용하는 것이 바람직하다.

5) 일반 선형근사 분석

가) 개요

일반 선형근사 분석법(Generalized LC)은 Eurocrypt'95에서Harpes-Kramer-Massey에 의해 제안된 분석법으로 반복 블록암호알고리즘에 적용 가능하다. Matsui가 제안한 LC는 선형근사를 사용한 반면에, 일반화된 LC는 I/O 합을 사용하여 키를 찾는 방법으로 LC를 확장한 공격법이다.

나) 개념

n-라운드의 선형근사식을 구성하기 위해서는 각각의 라운드에 대한 좋은 확률을 갖는 선형근사식들의 결합을 통해 이루어진다. 그런데 선형근사식을 일반적인 균형 함수를 사용하여도 되며, 다음과 같은 삼중합으로 표시할 수 있으면 선형근사 분석법과 동일하게 공격할 수 있다.

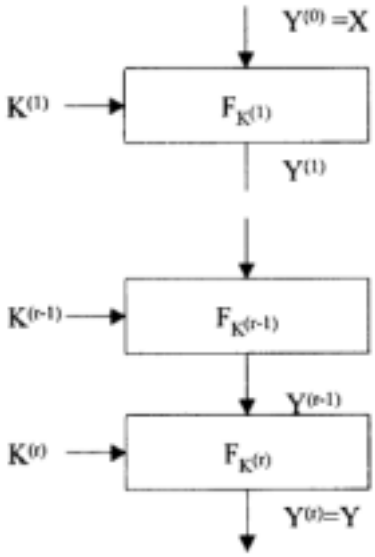
$$T^{(i)} = f_i(Y^{(i-1)}) \oplus g_i(Y^{(i)}) \oplus h_i(K^{(i)}), g_i = f_{i-1}$$

이 식이 성립할 확률이p라 하였을 때, Matsui는 bias=|1p-1/2|이 성공확률에 대한 중요한 요소가 됨을 사용하였으나, 일반 선형근사 분석을 제안한 Harpes는 이 개념을 확장시켜 imbalance의 개념을 도입하였다.

$$I(V) = |2P(V=0) - 1| = |E(2V-1)|$$

다) 공격 방법

먼저 일반화된 LC를 설명하기 위해 반복 블록 암호알고리즘에 사용된 기호를 살펴보자. i 번째 라운드의 입력을 $Y^{(i-1)}$, 출력을 $Y^{(i)}$, 라운드 키를 $Y^{(i)}$, 라운드 함수를 F_{K_i} 라고 쓴다. 또 평문을 X, 암호문을 Y라고 쓴다. 첫째 라운드의 입력은 $Y^{(i)}$이므로 $Y^{(0)} = X$이며, 블록 암호알고리즘의 전체 라운드 수를 r이라고 하면 r 라운드의 출력 $Y^{(r)}$은 암호문 Y이다.



I 라운드 I/O 합은 라운드의 입력 벡터 상에서 정의된 균형인 부울함수와 라운드의 출력 벡터 상에서 정의된 균형인 부울함수의 XOR이다. i 번째 라운드의 입력 $Y^{(i-1)}$ 상에서 정의된 균형인 부울함수를 f_i,출력 $Y^{(i)}$상에서 정의된 균형인 부울함수를 g_i라고 할 때 i번째 라운드의 I/O 합 $S^{(i)}$은 $F_i(Y_{i-1} \oplus g_i(Y^{(i)})$이다. 즉, $S^i = F_i(Y_{i-1} \oplus g_i(Y^{(i)})$이 다. 특히, $f(X) = a \cdot X (a \neq 0)$이고 $g(Y) = b \cdot Y (b \neq 0)$일 때 I/O 합은 Matsui가 LC 분석할 때 사용한 선형근사이다.

[예] 라운드 함수를 $F_K(X) = X \boxplus K$라고 하자. 여기서 $\boxplus$는 2^n법으로 한 덧셈이며, n은 블록 비트 수이다. 입력 벡터 X상에서 정의된 부울함수 f를 $MSB(X)$로 두자. 같은 방법으로 출력 Y상에 정의된 부울함수 g도라고 $MSB(Y)$두자. $MSB(X)$는 X의 최상위 비트(most significant bit)이다. 즉, $X = (x_{n-1}, \cdots, x_0)$이고 x_{n-1}을 최상위비트라고 하면, $MSB(X) = x_{n-1}$이다. 그러면 f와 g는 균형인 부울함수이다. 따라서 $f(X) \oplus g(Y) = MSB(X) \oplus MSB(Y)$는 1라운드 I/O 합이다.

ρ 개의 연속적인 I/O 합 $S^1, \cdots S^\rho$에서 마지막 라운드를 제외한 각 라운드의 출력함수와 그 다음 입력함수가 같은, 즉 $f_i = g_{i-1}$일 때 ρ개의 I/O 합은 연결 가능하다고 하며 그들의 합 $S^{1\cdots\rho}$는 $\oplus_{i=1}^{\rho} S^i = f_1(Y^{(0)} \oplus g_\rho(Y^{(\rho)})$로 쓸 수 있다. 이 합을 다중 라운드(ρ 라운드) I/O 합이라고 한다.

LC 복잡도를 계산하기 위해 선형근사식의 확률을 계산한다. ɣ 라운드 반복 블록암호알고리즘일 때 ɣ-1 라운드 선형근사가 필요하며, 이 선형근사의 확률을 p라고 할 때 LC 복잡도는 $\frac{8}{(p-0.5)^2}$ 정도이다. 선형근사 확률 p 가 0.5와 차이가 많을 수록 좋은 선형근사이며 LC 복잡도는 줄어든다. 반면에 일반화된 LC는 선형근사 대신 'imbalance'를 사용한다. imbalance는 0 과1 사이의 값으로 클수록 좋은 것이 된다.

[정의] 이진확률변수 V의 imbalance를 $I(V)$로 쓰며 | 2P(V=0)-1 | 로 정의한다. 즉, $I(V) = |2P(V=0)-1|$ 이다.

[주] $I(V) = |P-q| = |2p-1| = 2|p-0.5|$ 이므로 imbalance의 측도값은 선형근사식이 얼마나 좋은 지를 나타내는 측도값($|p-0.5|$)의 2배이다. 여기서 $p = P(V=0))$ 이며 $q = 1-p$ 이다.

[예] 라운드 함수를 $F_K(X) = X \boxplus K$, 블록 크기를 2비트(n=2), $V = MSB(X) \oplus MSB(Y) \oplus MSB(K)$라고 하자. 여기서 $Y - F_K(X)$이다. $X = (x_1, x_0), Y = (y_1, y_0)$, $K = (k_1, k_0)$라고 두면, $p = P(V=0) = (x_1 \oplus y_1 \oplus k_1 = 0) = 3/4$이다. 따라서 $I(V) - |2P(V=0)-1| = 1/2$이다.

이젠 키를 찾는 방법을 살펴보자. 일반화된 LC를 이용하여 r라운드 반복 블록 암호알고리즘의 키를 찾기 위해서는 좋은(imbalance가 큰) ɣ- 1 라운드 I/O 합 $S^{(1\cdots r-1)} = g_0(X) \oplus g_{r-1}(Y^{(r-1)})$을 찾아야 한다.

이 I/O 합을 이용하여 마지막 라운드의 서브키 K_r을 찾는 과정은 다음과 같다.

단계 0 : 마지막 라운드의 가능한 모든 키 k에 대해 카운트 c[k]를 설정하고 초기값을 0으로 둔다.

단계 1 : 평문/암호문 쌍 (x,y)을 선택한다.

단계 2 : 가능한 키 k에 대해, $y^{r-1} := F^{-1}(y)$를 계산한다.

$g_0(x) \oplus g_{r-1}(y^{r-1})$이면 카운트 c[$k$]에 1을 증가시킨다.

단계 3 : N 개의 평문/암호문 쌍에 대해 단계 1과 단계 2를 수행한다.

단계 4: | c[k] - $\frac{N}{2}$ | 이 최대가 되는 k가 마지막 라운드의 진짜 키로 간주한다.

이 공격을 적용하기 위해서는 선형근사식이 아니면서 위와 같은 특성을 만족하는 균형 함수를 찾아야 하는데 그러한 함수를 찾는 것은 매우 어려워 현재까지 이러한 개념을 이용하여 공격된 블록 암호알고리즘은 없다.

6) 분할 공격 / Mod n 분석

기본적인 선형근사 분석은 입력 비트와 출력 비트간의 선형 근사식을 이용하여 공격하는 방법으로 고려할 수 있는데, 이를 개념적으로 고려하면 입력 비트의 XOR를 '0'과 '1'의 두 부분으로 분할하였다고 볼 수 있다. 이를 입력과 출력을 몇 개의 부분으로 분할하여 입력의 특정 부분이 출력의 특정 부분으로 가장 많이 가는 특성을 이용하여 키를 찾는 방법을 분할 공격법(Partitioning Attack)이라 한다.

Mod n 공격이란 분할 공격법의 일종으로 모듈라 덧셈과 데이터의존 순환만을 사용한 RC5P에 적용된 Mod n 공격이다. 이 방법은 입력 블록을 Mod 3으로 나누어 통계적인 분포 특성이 균일하지 않음을 이용한 것이다. 또한 최근에 RC6도 입력의 하위 5비트가 0인 집합이 통계적으로 균일하게 암호화되지 않는 통계적인 특성을 이용하여 분석되었는데 이 역시 분할 공격법의 일종이라 할 수 있다.
분할 공격법은 데이터를 분할하는 방법이 매우 많기 때문에 효과적으로 분석할 수 있는 데이터 분할을 찾는 것이 공격의 핵심이다.

마. 기타

1) 보간 다항식 공격

블록 암호알고리즘에 사용되는 함수는 유한체 상의 다항식으로 볼 수 있는데 주어진 블록 암호알고리즘의 라운드 함수를 다항식으로 고려하였을 때 대수적 차수가 작은 경우 라운드 수를 반복하여도 대수적 차수가 그다지 커지지 않는다. 이러한 경우 전체 암호알고리즘을 다항식으로 고려하면 입력과 출력을 다항식의 점으로 볼 수 있으며, 다항식의 차수가 n인 경우 n+1개의 기지 평문으로 다항식을 구할 수 있다.
이와 같이 보간 다항식 방법으로 암호알고리즘을 분석하는 기법을 보간 다항식 공격이라 말한다. 보간 다항식 공격은 라운드 함수의 대수적 차수에 의존하기 때문에 고계 차분 공격과 유사하게 라운드 함수의 대수적 차수를 계산하는 소프트웨어만으로 충분히 안전성 평가를 할 수 있다.
그런데 대수적 차수가 가장 큰 경우는 유리다항식 관점에서는 대수적 차수가 그다지 크지 않을 수 있으며, 유리 다항식에 대한 보간법을 이용하면 동일한 분석이 가능한 바, 안전성 평가시 이점에 유의하여야 한다.

2) 차분선형근사 분석

가) 개요

DC와 LC를 동시에 이용하여 키를 찾는 선택평문 공격법으로 Crypto'94에서 Langford-Hellman에 의해 제안되었다. 확률이 1인 truncated differential과 좋은 선형근사를 찾아 키를 찾는 것으로 LC를 개선한 공격법이다. 이 분석법은 DES와 같은 반복 블록 암호알고리즘에 적용 가능하며 라운드 수가 작을 때 효과적이다.

8 라운드 DES일 때, 512개의 선택평문으로는 80% 확률로, 768개의 선택평문으로는 95% 확률로 10비트 키를 찾을 수 있다. DC와 LC 만을 이용한 것과 비교해 보면, Biham-Shamir는 DC를 이용하여 8 라운드 DES 키를 찾는데 5,000개의 선택 평문이 필요함을 보였다. 또 Matsui는 LC를 이용하여 8 라운드 DES 키를 찾는데 500,000개의 평문이 필요함을 보였다.

나) 적용방법

n라운드 블록 암호알고리즘의 DLC(Differential Linear Cryptanalysis) 방법은 다음과 같다.

1) 확률이 1인 r_1라운드 truncated differential을 찾는다.

- 라운드 수 ɣ 이 큰 truncated differential을 찾는다.

2) r_2($r_2 = n - r_1 - 1$)라운드 선형근사식을 찾는다.

- 선형근사의 입력은 truncated differential의 출력에만 의존하는 것으로 선택한다.

- 선형근사확률이 좋은(확률이 1/2과 차이가 많은) 선형근사를 선택한다.

위의 r_2라운드 선형근사식의 확률을 p라고 하면 DLC에 필요한 선택평문 수는 $\frac{8}{(0.5-(p^2+p^2))^2}$ 정도이다(단, $q=1p$ 이다).

Langford-Hellma은 8 라운드 DES에 대해 DCL 공격을 사용하였으나, 여기서는 설명을 간단히 하기 위해 7 라운드 DES로 한정한다. 먼저 기호를 정의하자.

· (L_i, R_i): i 라운드 출력

· (L_0, R_0): 평문

· (L_7, R_7): 암호문 (보통 (R_7, L_7)을 암호문으로 생각함)

· 번지는 왼쪽에서 1부터 64로 부여한다

(Matsui는 오른쪽에서 0부터 63까지 부여 함)

· S-box의 6개의 입력을 $(x_1, x_2, x_3, x_4, x_5, x_6)$으로 표시

(Matsui는 $(x_5, x_4, x_3, x_2, x_1, x_0)$으로 표시하였음)

· A[i]: A의 i 번째 비트

· $A[i,j,\cdots,k]=A[i]\oplus A[j]\oplus\cdots\oplus A[k]$

· DES의 초기치환과 역 초기치환은 무시함

이젠 확률이 1인 3 라운드 truncated differential과 3 라운드 선형근사를 찾는 과정을 살펴보자.

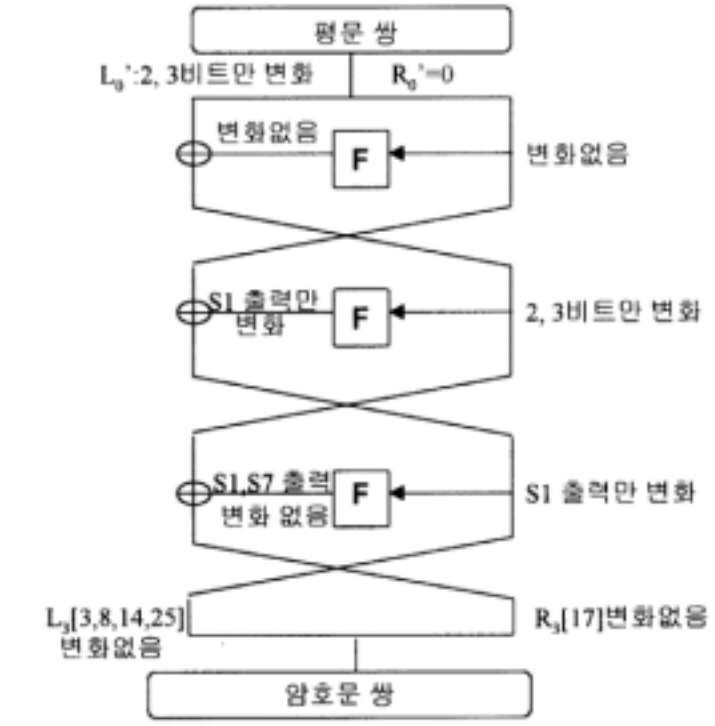
먼저 확률이 1인 3 라운드 truncated differential을 구한다(그림 참조).

- L'_0은 2, 3번째 비트만 제의하고 모두 '0'

- R'_0의 모든 비트는 '0', 즉 $R'_0=0$

- L'_3 의 3, 8, 14, 25 번째 비트만 선택

- R'_3 의 17비트만 선택



<확률1인 3 라운드 truncated differential>

즉, 평문의 오른쪽 부분은 같고, 왼쪽 부분도 2, 3비트를 제외하고는 같은 평문 쌍을 선택하면, 3 라운드 출력 중 왼쪽 3, 8, 14, 25 비트와 오른쪽 17비트는 항상(확률 1) 같다. 이 사실을 상세히 살펴보자. 1 라운드의 입력 XOR인 $R'_0 = 0$이므로 1 라운드 f 함수의 출력 XOR도 '0', 따라서 2 라운드 입력 XOR는 2, 3번째 비트만 제외하고 모두 0이다. 2, 3 번째 비트는 S1-box의 가운데 입력 비트로 작용하므로, 2 라운드 출력 XOR는 S1-box의 출력 4비트에 대응되는 것을 제외하고는 모두 '0'비트이다. 3 라운드의 입력 XOR는 1 라운드 입력 XOR와 2 라운드 출력 XOR와 XOR한 것이므로 3 라운드 입력 XOR는 S1-box의 출력 4비트에 대응되는 것을 제외하고는 모두 '0'비트이다.

S1-box의 출력 4비트에 대응되는 비트는 3 라운드의 S1-box와 S7-box에 영향을 주지 않는다. 따라서 3 라운드 f 함수의 출력 XOR 중S1-box와 S7-box의 출력에 대응되는 8 비트는 '0'이다. 따라서 $R_3[17]'$=0이다. 또 L'_3도 S1-box의 출력 4 비트에 대응되는 것을 제외하고는 모두 0이므로

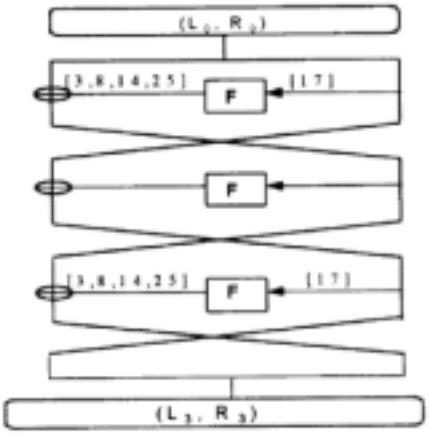
$L_3[3]' = L_3[8]' = L_3[14]' = L_3[25]' = 0$이다

이젠 3 라운드 선형근사를 살펴보자(그림 참조). 3 라운드 선형근사식으로 쓰면 다음과 같다.

$L_0[3,8,14,25] \oplus R_6[3,8,14,25] \oplus L_6[17] \oplus K_4[26] \oplus K_6[26]=0$이다.

한편 $R_6[3,8,14,25] = L_1[3,8,14,25] = CL[3,8,14,25]$이며, $L_6[17] = R_7[17] \oplus f(L_7, K_{7.1}[17] = CR[17] \oplus f(CL, K_{7.1})[17]$이므로 3 라운드 선형근사식은 다음과 같이 쓸 수 있다.

$$L_3[3,8,14,25] \oplus R_3[17] \oplus CL[3,8,14,25] \oplus CR[17] \oplus f(CL, K_{7.1})[17] \oplus K_4[26] \oplus K_6[26] = 0$$



<3라운드 선형근사>

이젠 키를 찾는 과정을 살펴보자. 확률 1인 truncated differential의 입력 XOR의 왼쪽 부분은 2, 3비트를 제외하고는 모두 '0'비트이고 오른쪽 부분은 모두 '0'비트이므로, 왼쪽 2, 3비트만 제외하고는 모두 같은 평문 쌍(x, $x*$)을 선택한다. 그러면 이 평문 쌍에 대응되는 2개의 선형근사식을 얻을 수 있다.

$$L_3[3,8,14,25] \oplus R_3[17] \oplus CL[3,8,14,25] \oplus CR[17] \oplus f(CL, K_{7,1})[17] \oplus K_4[26] \oplus K_6[26] = 0$$

$$L_3[3,8,14,25]^* \oplus R_3[17]^* \oplus CL[3,8,14,25]^* \oplus CR[17]^* \oplus f(CL^*, K_{7,1})[17] \oplus K_4[26] \oplus K_6[26] = 0$$

그런데 truncated differential의 확률은 1이고 출력 XOR 중 왼쪽 3, 8 ,14, 25 비트는 '0', 오른쪽 17비트는 '0'이므로, $L_3[3] = L_3[3]*$, $L_3[8] = L_3[8]*$, $L_3[14] = L_3[14]*$, $L_3[25] = L_3[25]*$이며, $R3[17] = R_3[17]*$이다. 이 사실을 이용하여 위의 2개의 선형근사식을 XOR하면

$$CL[3,8,14,25] \oplus CL[3,8,14,25]^* \oplus CR[17] \oplus CR[17]^* \oplus f(CL, K_{7,1})[17] \oplus f(CL^*, K_{7,1})[17] = 0$$

이다. 이 선형근사식의 확률은 $p^2 + p^2 = 0.576$이다. 왜냐하면 위의2개의 선형근사식의 확률은 각각 p 로, 동시에 성립할 수도 있고 동시에 성립하지 않을 수도 있기 때문이다. 이젠 선형공격과 같은 방법으로 7 라운드 서브키 중 S1-box에 대응되는 6비트 키 $K_{7.1}$을 찾을 수 있다. 이 때 필요한 평문/암호문의 개수는 $\frac{8}{(0.5-(p^2+q^2))^2}$ = 1358 쌍 이다. 그런데 이 보다 적은 암호문 쌍으로도 충분하다. 왜냐하면 임의 평문 P에 대해 왼쪽 2, 3 비트를 변화시켜 4개의 평문 $P_1 = P_1, P_2, P_3, P_4$를 얻을 수 있고, 대응되는 4개의 암호문 C_1, C_2, C_3, C_4 로부터 우리가 원하는 암호문 쌍 ${}_4C_2$ = 6개를 얻을 수 있기 때문이다.

바. 난수 통계특성법 정검

1) 도수 검정법

이진 수열 $x_1, x_2, \cdots, x_n$에 대해 0의 개수를 n_0, 1의 개수를 n_1이라고 하면 $n_0 + n_1 = n$을 만족한다. 도수 검정법은 수열 이 난수성을 만족하는 경우, 0과 1의 개수가 같다는 가정에 근거한 검정방법이다. 귀무가설은 '이진 수열이 랜덤하다' 이다. 귀무가설 하에서 한 비트가 0이 될 확률은 1/2, 1이 될 확률도 1/2이다. 귀무가설 하에서 0이 발생하는 기대도수 E_0는 n/2, 1이 발생하는 기대도수 E_1도 n/2이다. 0이 발생하는 실제 관측도수를 O_0, 1이 발생하는 실제 관측도수를 O_1이라고 하자.

x^2 적합도 검정을 적용해 보면 검정통계량 x^2값은 다음과 같다.

$$x^2 = \sum_{i=0}^{1} \frac{(O_i - E_i)^2}{E_i} = \sum_{i=0}^{1} \frac{2(O_i - \frac{n}{2})^2}{n}$$

$$= \frac{n}{2}\left[(\frac{O_0 - O_1}{2})^2 - (\frac{O_1 - O_0}{2})^2\right] = \frac{(n_0 - n_1)^2}{2}$$

위 통계량의 분포는 n이 클 때 자유도 1인 x^2 분포를 따르며 유의수준 α에 대한 기각역은 $x^2 > x^2(1, a)$ 이다

2) 도수-m 검정법

이진 수열 $x_1, x_2, \cdots, x_n$을 m비트 단위로 나누었을 때, $[\frac{n}{m}]$[17]개의 $(x_{km+1}, x_{km+2}, \cdots, x_{km+m})$, $k = 0, 1, 2, \cdots, n-1$들이 m차원에서 균등하게 분포되어 있는지를 검정하는 방법이다.

17) 여기서 $[x]$는 x보다 작지 않은 최소 정수를 의미한다.

이 때 n개의 m비트 블록들은 2^m개 중의 하나가 된다.
(0, 0, · · ·,0, 0)인 블록의 개수를 $n(0)$
(0, 0, · · ·,0, 1)인 블록의 개수를 $n(1)$
……
(1, 1,· · ·,1, 1)인 블록의 개수를 $n\,2^m-1$
이라 하자. 이진수열이 완전한 랜덤 수열이면 위의 2^m가지형태 블록의 각각의 확률은 $\frac{1}{2^m}$이므로 n개 중에서 위의 형태가 나올 평균 개수는 $\frac{n}{2^m}$이다. 그러므로 통계량

$$T=\sum_{i=0}^{2^m-1}\frac{\left(n(i)-\frac{n}{2^m}\right)^2}{\frac{n}{2^m}}=\frac{2^m}{n}\sum_{i=0}^{2^m-1}n(i)^2-n$$

은 근사적으로 자유도가 인 2^m-1인 x^2-분포를 따른다. 주어진 이진 수열이 랜덤하면, T의 값이 작은 값이 될 것이고, 랜덤하지 않다면 T의 값이 클 것이다. 따라서 유의수준α에 대한 기각역은 $t> x^2(2^m-1,a)$이다.
일반적으로 모든 m에 대하여 검정을 할 수 없으므로 사용되는 분야에 따라 적절히 m을 선택하여 사용한다. 메시지의 각 문자(character)를 비트의 집합으로 나타낼 경우에, 예를 들면, 영어의 알파벳 26을 비트의 수가 같은 것으로 나타내자면 적어도 $\log_2 26$비트를 사용해야 하므로 각 문자당 5 비트가 필요하다. ITA 2는 5-bit code이므로 우리는 m=5인 경우에 관심이 있다.
m=2일 때, 즉, 도수-2 검정은 계열검정으로 알려져 있기도 하다 (e.g. Knuth (1981) 60쪽).

$$T=\frac{2}{N}\sum_{i=0}^{1}\left(n(i)-\frac{N}{2}\right)^2$$

이 된다. 여기서 $n(i)$는 $i(i=0,1)$ 값을 갖는 비트의 수이다.

n=(0)+ n(1)을 이용하면 다음과 같이 된다.

$$T = \frac{(n(0) - n(1))^2}{n}$$

또 다른 검정통계량으로는 $Z = \frac{1}{\sqrt{\frac{n}{4}}}(n(1) - \frac{n}{2})$이 사용되며 Z는 근사적으로 표준 정규분포 N(0, 1)을 따른다. d진 수열에 대한 도수-m 검정법은 2진 수열의 경우와 마찬가지로 먼저 수열 { x_n}을 길이 m인 블록으로 나눈다. 그러면 각 d^m가지의 블록들의 각각의 확률은 $\frac{1}{d^m}$이 된다. $n_d(i)$를 i 번째 블록 형태의 개수라 하면 통계량

$$T = \frac{d^m}{n} \sum_{i=0}^{d^m-1} n_d(i)^2 - n$$

은 근사적으로 자유도가 d^m-1인 x^2 분포를 따른다.

3) Binary derivative 검정법

$x_1, x_2, \cdots, x_n$을 길이 n의 2진 수열이라고 하자. 0의 개수를 O_1이라고 하고 1의 개수를 O_2라 하자. 길이가 n인 비트 스트림의 binary derivative는 연속하는 두 비트를 modulo-two 덧셈하여 생성되는 새로운 길이 $n-1$의 비트 스트림을 말한다. 이때, 첫 번째 binary derivative를 $d_1(X)$라 한다. 첫 번째 binary derivative에서 도수 검정법을 이용하여 검정통계량 값 x_1^2을 구한다. 이와 같은 과정을 k번 반복하여 길이가 $n-k$인 k번째 binary derivative, 즉, $d_k(X)$를 계산한 후, 검정통계량 x_k^2을 구한다. 예를 들어, 10회 의 과정을 반복한다면 검정 통계량 x^2값은 $x^2 = mean(x_1^2, \cdots, x_{10}^2)$이 되며 도수 검정법을 통해 자료의 난수성을 검정한다.

4) Poker 검정법

도수-m 검정법은 크기 m인 블록을 겹치기 않게 나누어 각 블록의 가능한 패턴이 고른지를 검정하는 방법이다. 반면에 poker 검정법은 크기 m인 블록을 겹치기 않게 나누어 각 블록의 1의 개수만을 고려한 패턴이 고른지를 검정하는 방법이다. Porker 검정법을 hamming weight 검정법이라고도 한다. 도수-m 검정에서 사용한 통계량 보다 단순한 다음의 통계량을 사용하면 공식이 간단하여 쉽게 이용할 수 있다. 그러나 도수--m 검정에서 사용한 통계량보다는 많은 정보의 누수가 발생하므로 정밀성은 도수-m 검정 통계량을 사용한 것보다 떨어진다.

도수-m 검정법과 같이 수열을 크기가 m인 $[\frac{n}{m}]$개의 블록으로 나눈다. poker 검정에서는) $n(i)(0 \le i \le m)$를 i개의 1과 $m-i$개의 0으로 구성된 블록의 개수라 하자. 수열이 랜덤하면 $n(i)$의 평균은 $\overline{n(i)} = \binom{m}{i}, 0 \le i \le m$이다. 그러므로

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

는 근사적으로 자유도가 m인 x^2-분포를 따른다.

5) Runs 검정법

(1) 런의 수 검정법

n개의 이진 수열 $x_1, x_2, \cdots, x_n$에 대해 런(runs)을 고려해보자. 런이란 연속적으로 같은 값이 나열된 부분수열로, 1로 구성된 런을 'block', 0으로 구성된 런을 'gap'이라고 한다. 예를 들어 이진 수열이 '00111011'일 때 'block'은 '111' '11'이며 'gap'은 '00' '0'으로, block 길이가 2인 것이 1개 3인 것이 1개, gap 길이가 2인 것이 1개, 1인 것이 1개이다.

새로운 런의 시작은 'change'라고 하면 위의 예에서 change는 3번 일어났으며 런의 개수는 4이다. 런의 개수는 change 수에다 1을 더하면 된다. 즉 rums의 수 =change의 수 + 1 이다.

이진 수열이 랜덤하다고 가정하면 change가 일어날 확률은 1/2이며, n개의 이진 수열에서 change의 개수는 평균이 $(n-1)/2$, 분산이 $(n-1)\,2^{2}$이항분포를 따른다. 런의 수(block의 수 + gap의 수)를 R이라고 하면 R의 확률분포는 change 수의 분포를 이용하여 구할 수 있다.

$$\begin{aligned} P(R=i) &= P(C+1=i) = P(C=i-1) \\ &= \binom{n-1}{i-1}(1/2)^{i-1}(1/2)^{n-1-(i-1)} \\ &= \binom{n-1}{i-1}2^{1-n} \end{aligned}$$

여기서, C는 change 수이며 $C \sim b(n-1,\ 1/2)$인 이항분포이다. 런의 수 R의 확률분포를 이용하여 런의 개수의 평균 $E[R]$과 분산 $Var(R)$을 계산할 수 있다.

$$E[R] = \sum_{i=1}^{n} i\ P(R=i) = \sum_{i=1}^{n} i \binom{n-1}{i-1}2^{1-n} = \frac{n+1}{2}$$

$$\begin{aligned} Var(R) = E[R^2] - (E[R])^2 &= \sum_{i=1}^{n} i^2\ P(R=i) - (\frac{n-1}{2})^2 \\ &= \sum_{i=1}^{n} i^2 \binom{n-1}{i-1}2^{1-n} - (\frac{n-1}{2})^2 \\ &= \frac{n-1}{4} \end{aligned}$$

n 크면 R은 정규분포로 근사 시킬 수 있다.

$$\frac{R-E[R]}{\sqrt{Var(R)}} = \frac{R-\frac{n+1}{2}}{\sqrt{\frac{n-1}{4}}} = \frac{2R-n-1}{\sqrt{n-1}} \sim N(0,1)$$

따라서 검정통계량 z를 $\frac{2R-n-1}{\sqrt{n-1}}$ 으로 두면 귀무가설(이진수열이 랜덤) 하에서 z 는 표준정규분포를 따르며 유의수준 α에 대한 기각역은 z 또는 $z < z_{a/2}$ 또는 $z < -z_{a/2}$이다.

여기서 $P(Z> z_a)=a$이며 Z는 표준정규 확률변수이다. 또 길이 이진 수열을 S(S는 n비트 블록 개수)개라고 하면 런의 개수가 i인 평균 블록 개수 e_i는 다음과 같다.

$$e_i=\binom{n-1}{i-1}2^{1-n}\times S$$

S개의 블록 중 런의 개수가 i인 블록 개수의 관측값을 o_i라고 하고, 검정 통계량을 $Q=\sum_{i=1}^{n}\frac{(o_i-e_i)^2}{e_i}$로 두면, 귀무가설 하에서 Q는 자유도 $n-1$인 x^2분포를 가지며 유의수준 α에 대한 기각역은 $q> x_a^2(n-1)$이다.

(2) 런의 길이 검정법

런의 수뿐만 아니라 런의 길이를 조사하는 것은 중요하다. $x_1, x_2, \cdots, x_n$을 길이가 n인 2진 수열이라 할 때. 0 혹은 1이 연속하여 나타나는 정도가 추정치에 근접한 지를 확인하는 검정방법이다. 길이가 I인 run에 대하여 1의 run(block)의 개수와 0의 run(gap)의 개수를 각각 B_i, G_i라고 하면, 길이가 i 인 block(gap)의 개수의 기대값 E_i는 다음과 같다.

$$E_i=\begin{cases}\dfrac{\frac{n-i-1}{2}+2}{2^{i+1}} & , i=1,2,\cdots,n-1\\ \dfrac{1}{2^n} & , i=n\end{cases}$$

검정 검정통계량을 다음과 같이 정의하자.

$$x^2=\sum_{i=1}^{k}\frac{(B_i-E_i)^2}{E_i}+\sum_{i=1}^{k}\frac{(G_i-E_i)^2}{E_i}$$

그러면 검정 검정통계량 x^2는 자유도가 $2k-2$인 x^2분포를 따른다.

6) 계열(Serial) 검정법

계열 검정법은 이진 수열 $X_1, X_2, \cdots, X_n$에서 한 비트가 그 다음 비트로 전이되는 과정이 랜덤한지를 조사하기 위한 검정법이다. 이진 수열을 연속된 2 비트 $X_1, X_2, X_3, X_4, \cdots, X_{n-1}, X_n$으로 나누었을 때 "00"의 개수를 n_{00}, "11"의 개수를 n_{11}이라 하고 전체 n비트 중 "0"의 개수를 n_0, "0"의 개수를 n_1하자. 그러면 다음의 식을 얻는다.

$$n_{00} + n_{01} = n_0 \text{ 혹은 } n_0 - 1$$
$$n_{10} + n_{11} = n_1 \text{ 혹은 } n_1 - 1$$
$$n_{00} + n_{01} + n_{10} + n_{11} = n - 1$$
$$n_0 + n_1 = n$$

위 식에서 -1이 나타내는 이유는 길이 n개의 부분에서는 단지 n-1개의 전이가 일어나기 때문이다. n_{ij}의 기대치는 $\frac{n-1}{4}$ 이므로 다음의 통계량은 근사적으로 자유도 2인 x^2-분포를 따른다.

$$T = \sum_{i,j=0}^{1} \frac{(n_{ij} - \frac{n-1}{4})^2}{\frac{n-1}{4}} - \sum_{i=0}^{1} \frac{(n_i - \frac{n}{2})^2}{\frac{n}{2}}$$

이것을 다른 식으로 표현하면

$$T = \frac{4}{n-1}(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2) - \frac{2}{n}(n_0^2 + n_1^2) + 1$$

로 간단한 식이 된다. 계열검정은 계열상관검정(serial correlation test)으로 알려져 있기도 한다.(Kunth (1981), 20쪽)

일반적으로 길이가 b인 부분수열을 겹치게 생성하여 2^b개의 부분수열에 대한 형태의 균일성을 조사하는 검정방법이다.

부분수열($x_1, x_2, \cdots, x_b$)는 $t = \sum_{i=1}^{b} 2^i x_i, 0 \le t \le 2^b - 1$로 표현될 수 있으며, n_t를 임의의 부분수열에서 발생하는 10진수의 값 t의 개수라고 하자. 길이가 b인 부분수열을 갖는 이진 수열의 균일성에 대한 검정 통계량을 다음과 같이 정의한다.

$\Psi_b^2 = \frac{2^b}{n-b+1} \sum_{t=0}^{2^b-1} \left(n_t - \frac{n-b+1}{2^b} \right)^2$, 여기서 $\Psi_b^2 = 0$이다.

이제 0의 개수를 n_0,1의 개수를 n_1이라고 하자. b=2라고 했을 때, 즉, 부분수열 $(x_1, x_2), (x_2, x_3), \cdots, (x_{n-1}, x_n)$에서 발생하는 형태는 00, 01, 10, 11이며 각각의 개수를 $n_{00}, n_{01}, n_{10}, n_{11}$이라고 한다. 귀무가설은 '각각의 형태에서 발생하는 개수가 같다'이다. 따라서 검정 통계량 Ψ_2^2(b=2일 때 Ψ_b^2)는 다음과 같다.

$$\begin{aligned}\psi_2^2 &= \frac{4}{n-1} \sum_{i=0}^{3} \left(n_i - \frac{n-1}{4}\right)^2 \\ &= \frac{4}{n-1}\left[\left(n_0 - \frac{n-1}{4}\right)^2 + \left(n_1 - \frac{n-1}{4}\right)^2 + \right. \\ &\qquad \left. \left(n_2 - \frac{n-1}{4}\right)^2 + \left(n_3 - \frac{n-1}{4}\right)^2\right]\end{aligned}$$

여기서 부분수열의 형태는 00, 01, 10, 11이므로 $n_0 = n_{00}$, $n_1 = n_{01}$, $n_2 = n_{10}$, $n_3 = n_{11}$이라 두면

$$\begin{aligned}\psi_2^2 &= \frac{4}{n-1}\left[\left(n_0 - \frac{n-1}{4}\right)^2 + \left(n_1 - \frac{n-1}{4}\right)^2 + \right. \\ &\qquad \left. \left(n_2 - \frac{n-1}{4}\right)^2 + \left(n_3 - \frac{n-1}{4}\right)^2\right]\end{aligned}$$

이다. 또한 다른 통계량 는 Ψ_{b-1}^2는

$$\psi_{b-1}^2 = \frac{2^{b-1}}{n-(b-1)+1} \sum_{t=0}^{2^{b-1}-1} \left(n_t - \frac{n-(b-1)+1}{2^{b-1}}\right)^2$$

이다. b = 2를 대입하면

$$\psi_1^2 = \frac{2}{n} \sum_{t=0}^{1} \left(n_t - \frac{n}{2}\right)^2$$

$$= \frac{2}{n}\left[(n_0 - \frac{n}{2})^2 + (n_1 - \frac{n}{2})^2\right]$$

이다. 즉, Ψ_1^2은 도수 검정 통계량이다. 위의 Ψ_2^2와 Ψ_1^2에 대한 통계량 Ψ^2를 $\Psi^2 = \Psi_2^2 - \Psi_1^2$로 두면

$$\psi^2 = \frac{4}{n-1}(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2) - \frac{2(n_0^2 + n_1^2)}{n} + 1$$

이다. Ψ^2은 $n \geq 21$일 때 자유도가 2인 X^2 분포를 따른다.

7) 자기상관(Autocorrelation) 검정법

계열검정은 바로 뒤에 있는 비트로 전이되어 가는 과정을 조사한 것이다. 이와 같은 과정을 d비트 떨어진 비트간의 전이를 생각한 것이 자동상관검정이다. 이제까지 알려진 자동상관검정에서 검정통계량들은 다음 4가지가 있다.

① $(X_1, X_{d+1}), (X_2, X_{d+2}), \cdots, (X_{n-d}, X_n)$, 로 이루어진 2비트들에 대하여 계열검정과 마찬가지로 n_{ij}를 구하여 검정하는 것이 자동상관검정이다.

② 이진수열 n비트 $X_1, X_2, \cdots, X_n$으로부터 $(X_1, X_2, \cdots, X_{n-d})$와 $(X_{d+1}, X_{d+2}, \cdots, X_n)$ 와의 상관관계를 조사하는 통계량으로 다음을 생각하여 보자.

$$A(d) = \sum_{i=1}^{n-d}(1 - 2x_i)(1 - 2x_{d+i})$$

이진수열이 랜덤하면, 즉, 독립이고 같은 분포를 갖으면, 통계량 $A(d)$의 평균은 0이고 분산은 $n-d$이다. $n-d$가 충분히 클 경우 (예를 들면 $n-d \geq 30$)에는 중심 극한 정리에 의하여 $Z = \frac{A(d)}{\sqrt{n-d}}$ 는 근사적으로 표준 정규분포 $N(0, 1)$을 따른다.

③ 또 다른 자동상관 검정은 다음의 통계량을 사용한다.

$$B(d) = \sum_{i=1}^{n-d} x_i x_{i+d}, \qquad 1 \le d \le n$$

만약 이진 수열 $x_1, x_2, \cdots, x_n$이 n_0개의 0을 n_1개의 1을 갖고 있으면 x_i, x_{i+d}의 기대치는 $(\frac{n^i}{n})^2$이고 $B(d)$의 기대치는 $(\frac{n_1}{n})(n-d)$이다.

④ 일부학자들은 다음의 검정을 자동상관검정이라고도 한다. $x_1, x_2, \cdots, x_n$으로 부터 다음과 같은 새로운 수열을 만든다.

$$x_1 \oplus x_{d+1}, x_2 \oplus x_{d+2}, \cdots, x_{n-d} \oplus x_n$$

이때 새로 구성된 수열에 관한 도수-1 검정을 원래의 수열에 관한 delay가 d인 자동상관이라고 한다.

8) Change Point 검정법

$x_1, x_2, \cdots, x_n$을 길이가 n인 이진 수열이라고 하자. Change point 검정은 비트 스트림에서 t번째 비트에 대하여 그 이전 1의 비율과 그 이후의 1의 비율을 비교하여 그 차이가 최대인 점을 찾는 방법이다. 수열이 랜덤하면 1의 비율에 변화가 없다. $S[n]$을 전체 1의 개수, $S[t]$를 t비트까지의 1의 개수라고 할 때, 검정 통계량 $U[t]$를 $U[t]=nS[t]-tS[n]$이라고 정의한다. 만약 t비트까지의 1의 비율이 전체에서의 1의 비율보다 작다면, 즉 $\frac{S[t]}{t} < \frac{S[n]}{n}$이면, $U[t]<0$이다. t비트까지의 1의 비율이 전체의 1의 비율보다 크다면, 즉 $\frac{S[t]}{t} > \frac{S[n]}{n}$이면, $U[t]>0$이다. 여기에서 0과 1의 개수를 바꾸었을 때 두 값은 크기는 같고 부호만 바뀔 것이다.

따라서 1의 비율에 있어 증가나 감소에는 관계없이 크기가 가장 많이 변화하는 점을 찾는 것이 목적이므로 $U[t]$의 절대값 중 최대인 점, 즉 $M=\max|U[t]|, i=1,2,\cdots,n-1$, 으로 결정한다. $U[t]$의 절대값 중 최대인 것을 M이라면, 즉, $M=\max|U[t]|$이면, p-value값 α는 근사적으로 $a=e^{-\frac{2M^2}{nS[n](n-S[n]}}$ 이다.

9) 충돌(Collision) 검정법

도수-m 검정에서와 마찬가지로 충돌검정에서도 이진 수열 $x_1, x_2, \cdots, x_n$을 m비트씩 분할하여 $[\frac{n}{m}]$개의 블록을 만든다. 도수-m 검정과 충돌 검정과의 차이점은 도수-m 검정에서는 m을 작은 수로 취하지만 충돌검정에서는 m을 큰 값으로 하며 $[\frac{n}{m}]<<2^m$이 되도록 하여 범주의 수 2^m이 관찰치의 수 $[\frac{n}{m}]$보다 클 때 사용하는 방법으로 부분수열 $x_{mj+1},\cdots, u_{mj+m}$이 m차원 공간에서 균등하게 분포되어 있는가를 알아보기 위한 검정방법이다.

10) Cuppon collector's 검정법

d-진 수열 $x_1, x_2, \cdots, x_n$이 독립이고 {0, 1, …, d-1} 상에 같은 분포를 갖는지를 검정하기 위하여 0부터 d-1 까지 모든 정수를 처음으로 포함하게 되는 부분 정수열 $x_{j+1}, x_{j+2}\cdots, x_{j+r}$ 의 길이 r의 분포를 계산하여 x^2-검정을 이용하는 것이다. 이 정법의 이름은 다음과 같은 실제적인 문제로부터 나온 것이다. d종류의 쿠폰 중 1개씩 과자봉지에 들어있을 때 d 종류의 쿠폰 모두를 수집할 때까지의 과자봉지의 수가 r이 된다.

d-진 수열의 n비트 $x_1, x_2, \cdots, x_n$이 주어졌을 때 0부터 d-1까지 모든 정수를 포함하게 되는 최소의 r 비트 $x_{j+1}, x_{j+2} \cdots, x_{j+r}$의 개수가 적어도 5이상이 되도록 t를 결정하고 $r = d, \cdots, t$의 $t - d + 1$개의 범주로 분류한다. 이때 각 확률은 다음과 같다.

$$P_r = \frac{d!}{d^r} S(r-1, d-1), \quad d \geq r < t$$

$$P_t = 1 - \frac{d!}{d^{t-1}} S(t-1, d)$$

단, $S(n, i)$는 제 2종의 Stirling 수이다. 위의 자료를 이용하여 x^2-검정을 할 수 있다. 이진 수열인 경우에는 수 0,1 대신에 m비트로 이루어진 2^m개의 (0, 0, …, 0),(1, 0, …,0), …,(1, 1, … ,1)로 대체하여 모든 $d = 2^m$의 블록을 포함되게 하는 최소의 r 비트를 생각하여 위와 마찬가지로 x^2-검정을 할 수 있다. 이 경우에는

$$P_r = \sum_{k=1}^{d-1} C(d-1, k) \left(\frac{k}{d}\right)^{r-1} (-1)^{k+1},$$

$$P_t = \sum_{k=1}^{d-1} C(d, k) \left(\frac{k}{d}\right)^{t-1} (-1)^{k+1},$$

단, C(n, i)는 이항계수로 간단히 나타난다.

11) Gap 검정법

Gap 검정법은 이진 수열이 아닌 [0, 1) 상의 값을 갖는 수열의 랜덤성을 검정하는 방법이다. (x_n)가 $i.i.d.$이고 X가 [0, 1)상의 일양분포를 가질 때, 즉, $f(x) = x$,(X_n)이 random 한지를 검정하는 방법이다.

임의 주어진 구간(α,β), 0≤α,β≤1에 대해 연속적 부분 수열 $X_j, X_{j+1}\cdots, X_{j+r}$로써 $X_{j+r}\in[a,B)$이고 다른 x들은 [α,β)에 속하지 않을 때 이 연속적 부분 수열의 gap의 길이를 r이라고 한다. $(X_n)=i.i.d.$이므로 Gap 길이의 분포는 모수가 $p=P(a<X<B)=b-a$인 기하분포를 따른다. [0, 1) sequence $\{X_n\}$를 관찰한 후 Gap의 길이가 0, 1, …, t-1가 t이상인 경우를 생각하여 t + 1개의 범주로 구분한다. 이 때 n과 t는 각 범주의 개수가 5이상이 되도록 선택한다. 그러면 Gap 길이다 각 범주에 속할 확률은 $P_0=p$, $P_1=p(1-p)$, $P_{t-1}=p(1-p)^{t-1}$, $P_t=1-p^t$이다. 따라서 자유도가 t인 x^2-검정을 적용할 수 있다. 특히 $(a,B)=(0,\frac{1}{2})$ 또는 $(\frac{1}{2},1)$인 경우가 보통 runs above the mean 또는 runs below the mean 이라고 불리는 검정법이다.

12) 순열(Permutation) 검정법

순열 검정법은 이진 수열이 아닌 [0, 1) 상의 값을 갖는 수열의 랜덤성을 검정하는 방법이다. [O, 1)수열 $(u_0, u_1, u_2,\cdots)$을 m비트씩 m개의 군으로 나눈다. 즉, $<u_{mj}, u_{mj+1},\cdots, u_{mj+1-1}>$, $0\le j\le n$,이 n개의 m비트들을 다음의 $m!$ 개의 가능한 상대순서로 된 범주에 속하는 개수를 계산한다.

① $u_{mj} < u_{mj+1} < \cdots < u_{mj+m-2} < u_{mj+m-1}$

② $u_{mj} < u_{mj+1} < \cdots < u_{mj+m-1} < u_{mj+m-2}\cdots$

$(m!)$ $u_{mj+1} < u_{mj+m-2} < \cdots < u_{mjj+1} < u_{mj}$

그리고 (u_n)가 로부터 나온 i. i. d. 실험치이며 각 범주에 속할 확률은 $\frac{1}{m!}$ 이 되어서 자유도 $m!-1$인 x^2 검정을 적용할 수 있다.

13) 최대값(Maximum) 검정법

최대값 검정법은 이진 수열이 아닌 [0, 1) 상의 값을 갖는 수열의 랜덤성을 검정하는 방법이다. [O, 1) 수열 ($u_0, u_1, u_2, \cdots$)에 대해 $u_j = \max < u_{mj}, u_{mj+1}, \cdots, u_{mj+1-1} >$, $0 \le j \le n$을 이용하여 [0, 1) 내에 값을 취하여 수열이 독립이고 [0, 1)상에서 균등하게 분포되어 있는가를 검정하는 방법이다.

만약 (X_n)가 $i.i.d.$ 이고 [0, 1)상에서 균등하게 분포되어 있다면 $V_j = \max(u_{mj}, u_{mj+1}, \cdots, u_{mj+1-1})$의 분포는 $F(v) = P(V_j \le v) = P(x_{mj} \le v, \cdots, x_{mj+m-1} \le v) = v^m$이 된다. 따라서 Kolmogorov-Smirmov 검정통계량 $D = | F_n(v) - F(v) |$ 을 이용하여 (u_n)가 독립이고 [0, 1)내에 균등하게 분포되어 있는지 검정할 수 있다.

14) 엔트로피 검정법

난수 생성기에 대한 Maurer의 엔트로피 검정은 기존의 통계적 관점의 검정에 비해 좀더 일반적인 통계적 모델을 기반으로 한다. Maurer의 엔트로피 검정은 기존에 사용되는 통계적 검정(즉, 빈도검정, 계열 검정, 포커 검정, 런 검정, 자기상관 검정을 포함하는)이외에 다음과 같은 두 가지 주된 이점을 제공한다. 첫째로, 특정한 유형의 통계적 결점을 찾는 기존의 통계적 검정들과는 달리, Maurer에 의한 엔트로피 검정은 난수 발생기들이 가질 수 있는 발생 가능한 결점들에 대한 매우 일반적인 class 중의 하나를 찾을 수 있게 한다.

이러한 결점들에 관한 class는 유한 메모리를 가지는 ergodic stationary source에 의해 모델화 될 수 있는데, 이는 난수 비트 발생기의 실제적인 구현에서 발생 가능한 결점들을 포함하도록 적절하게 나타내 질 수 있다. 둘째로, Maurer에 의한 엔트로피 검정은 난수 발생기의 결점에 관한 실제적인 암호학적 중요성을 측정한다. 좀더 정확하게 말하자면, 검정 매개변수는 키 소스의 비트당 엔트로피를 측정하는데, 이는 공격자가 비밀 키 소스의 통계적 결점에 관한 지식을 부당하게 이용할 때 공격자가 수행하는 최적키 탐색 전략의 실행 시간과 관련이 있다. 즉, 비밀 키 소스의 비트당 엔트로피는 암호 시스템을 깨기 위해 전 탐색보다 더 빠른 방법이 존재하지 않는다는 가정 하에서 암호 시스템의 효율적인 키 크기를 측정한다.

이러한 두 가지 이점은 유한 메모리 $M \le L$(L : 검정 파라미터)을 갖는 이진 ergodic stationary source의 일반화된 class와 소스의 비트 당 엔트로피 H와 밀접하게 관련된 통계량 T를 결과로 가지는 조건부 확률 모델을 설정했다는 사실에 기인한다.

검정 통계치 T는 양의 정수 값인 파라미터 L, Q, K에 의해 설정된다. 검정을 수행하기 위해, 먼저 $s^N = s_o, s_1, \cdots, s_{N-1}$을 전체의 길이가 N인 이진 수열이라 두고 s^N에서 길이 L의 겹치지 않는 인접한 블록을 만든다. 수열 s^N의 전체 길이는 $N = (Q+K)L$이 되는데, 여기서 K는 검정 단계의 블록의 수이고 Q는 초기화를 위한 블록의 수이다.

<검정 알고리즘>

단계 1 : 처음 Q개의 블록을 초기화한다.

단계 2 : $K = \frac{N}{L} - Q$개의 검정 블록에 대해 다음을 수행한다.

$b_n(s^N) = [s_{Ln}, s_{Ln+1}, \cdots, s_{Ln+l-1}]$을 n번째 블록이라 하고, $Q \le n \le Q + K - 1$에 대하여 다음을 정의한다.

$$A_n(s^N) = \begin{cases} \min\{i \mid 1 \le i \le n, b_n(s^N) = b_{n-i}(s^N)\} & \text{if } \{\cdots\} \neq \phi \\ n & \text{if } \{\cdots\} = \phi \end{cases}$$

즉, $A_n(s^N)$는 n번째 블록 $b_n(s^N)$과 처음으로 일치하게 되는 $b_{n-i}(s^N)$가 존재할 때 i로 정의하며, 존재하지 않는 경우 n으로 정의한다.

단계 3 : 단계 2에서 얻어지는 결과 값을 이용하여 다음과 같이 정의되는 통계량`를 계산한다.

$$T(s^N) = \frac{1}{K} \sum_{n=Q}^{Q+K-1} \log_2 A_n(s^N)$$

검정 통계치 T의 구현을 위해 파라미터 L, Q, K에 대해 다음의 값을 적용하고 있다. $1 \le L \le 12$, $Q \ge 30 \cdot 2^L$ 그리고 K는 클수록 좋다(예를 들면, $K=1000 \cdot 2^L$). 이러한 Q의 선택은 거의 확률 1로써 난수열에서 모든 L-비트 형태가 최소한 한 번씩 처음부터 Q블록 내에 나타나도록 한 것이다. $T(U^n{}_{BMS_{\frac{1}{2}}})$의 평균과 분산은 다음 표와 같다.

L	$E(T(U^N))$	$K \cdot Var(T(U^N))$	L	$E(T(U^N))$	$K \cdot Var(T(U^N))$
1	0.73264948	0.68977	7	6.19625065	3.12539
2	1.53743829	1.33774	8	7.18366555	3.23866
3	2.40160681	1.90133	9	8.17642476	3.31120
4	3.31122472	0.35774	10	9.17232431	3.35646
5	4.25342659	2.70455	11	10.1700323	3.38409
6	5.21770525	2.95403	12	11.1687649	3.40065

위의 표에서는 L, Q- $\cdot \infty$, K를 갖는 검정 통계량 $T(U^n{}_{BMS_{\frac{1}{2}}})$의 평균과 분산 값이 주어진다. 이때, U^N은 실제 난수열이고 소스가 $BMS_{\frac{1}{2}}$(Biased Memory-less Source)인 경우이다. 이는 난수 발생기의 소스가 $BMS_{\frac{1}{2}}$이라는 가설검정에 이용된다.

또한 $Q \to \infty$를 가정한 것은 실제의 검정에서 Q를 충분히 크게 선택하므로 오차가 아주 작은 것으로 볼 수 있어서 실제로 이용하는데 문제가 없다.

15) 선형 복잡도(Linear Complexity) 검정법

수열 s^n을 $s_0, s_1, \cdots, s_{n-1}$의 항들을 갖는 길이 n의 유한 이진수열이라 둔다면, 선형 복잡도 검정은 수열 s^n의 선형 복잡도를 결정하기 위한 검정법이다. 이는 LFSR(Linear Feedback Shift Register)를 분석하는 데에 유용하게 적용할 수 있다.

[정의] 선형 복잡도(LC)

$L(s^n)$으로 표기되는 유한 이진수열 s^n의 선형 복잡도는 처음 n항으로서 s^n을 갖는 수열을 생성시키는 가장 짧은 LFSR의 길이를 나타낸다.

[정의] Next discrepancy d_N

$s^{N+1} = s_0, s_1, \cdots, s_{N-1}, s_N$인 유한 이진수열을 고려하자. $C(D) = 1 + C_1 D + \cdots + C_1 D^L$에 대해, $<L, C(D)>$를 부분수열 $s^N = s_0, s_1, \cdots, s_{N-1}$을 생성시키는 LFSR이라고 둔다. Next discrepancy d_N은 LFSR에 의해 생성되는 s_N과 $(N+1)$번 째 항간의 차이(difference)가 되고 다음과 같이 나타낸다

$$d_N = (s_N + \sum_{i=1}^{L} c_i s_{N-i}) \bmod 2$$

$s^N = s_0, s_1, \cdots, s_{N-1}$을 선형 복잡도 $L = L(s^N)$을 갖는 유한 이진수열이라 하고, $<L, C(D)>$을 s^N을 생성하는 LFSR이라 둔다.

① LFSR $<L, C(D)>$가 $s^{N+1} = s_0, s_1, \cdots, s_{N-1}, s_N$을 생성한다는 것은 곧, next discrepancy d_N이 0이라는 것과 동치이다.

② 만일 $d_N = 0$이라면, $L(s^{N+1}) = L$이다.

③ $d_N = 1$이라 가정한다면, N보다 작은 가장 큰 정수 m에 대하여 $L(s^m) < L(s^N)$이 성립한다. 그리고 $<L(s^m) < B(D)>$을 s^m을 생성시키는 길이 $L(s^m)$의 LFSR이라 둔다면, $< L', C'(D)>$는 s^{N+1}을 생성시키는 가장 작은 길이의 LFSR이 된다.

여기서, $C'(D) = C(D) + B(D)\, D^{N-m}$이고 $L' = \begin{cases} L, & \text{if } L \le N/2 \\ L, + 1 - L, & \text{if } L \le N/2 \end{cases}$ 이 된다.

유한 이진수열 s^n의 선형 복잡도를 결정하기 위한 효율적 알고리즘인 Berlekamp-Messey 알고리즘을 아래에서 소개한다.

<검정 알고리즘>

입력 : 길이 n의 이진수열 $s^n = s_0, s_1, \cdots, s_{n-1}$

출력 : s^n의 선형 복잡도 $<L(s^m), 0 \le L(s^m) \le n.$

단계 1 : 초기화. $C(D) \leftarrow 1, L \leftarrow 0, m \leftarrow -1, B(D) \leftarrow 1, n \leftarrow 0$

단계 2: $N < n$인 동안 다음의 과정을 반복한다.

2.1 next discrepancy d를 계산한다.

2.2 만일 $d = 1$이면 다음을 수행한다.

$$T(D) \leftarrow C(D), \quad C'(D) = C(D) + B(D) D^{N-m}$$

만일 $L \le N/2$이면 $L \leftarrow N + 1 - l, m \leftarrow N, B(D) \leftarrow T(D)$

2.3 $N \leftarrow N + 1$

단계 3 : L값을 출력한다.

16) 선형 복잡도 프로파일(Linear Complexity Profile) 검정법

선형 복잡도 프로파일(LCP)은 1986년 Rainer A. Rueppel이 개념을 도입한 이래로 많은 학자들의 연구의 대상으로 각광받고 있으며 실제 Stream 암호 시스템의 비도를 결정하는데 가장 중요한 역할을 하는 요소 중의 하나이다.

[정의] 선형 복잡도 프로파일(LCP)

$L_n(s)$를 유한 이진수열 $s^n = s_0, s_1, \cdots, s_{n-1}$의 선형 복잡도(LC)라 할 때, 수열 $s = s_0, s_1, \cdots, s_{n-1}, \cdots$에 대해 수열 $L_1(s), L_1(s), \cdots$를 s의 선형 복잡도 프로파일(LCP)이라 한다.

정의에 의해 선형 복잡도 프로파일은 양의 수로 이루어진 Non-decreasing 수열이고 언급한 Berlekamp-Massey 알고리즘에 의하여 쉽게 계산될 수 있다. 선형 복잡도 프로파일의 개념이 나타나게 된 동기중의 하나는 어떤 이진 수열의 선형 복잡도(LC)가 크다고 해서 반드시 그 수열이 임의성을 갖는다고 볼 수 없기 때문이다.

선형 복잡도 프로파일(LCP)의 특성은 다음과 같다.

$L_1, L_1, \cdots$를 수열 $s = s_0, s_1, \cdots$의 LCP라고 두자.

① 만일 $j > i$이면, $L_j \geq L_i$이다.

② $L_N \leq N/2$일 때만, $L_{N+1} > L_N$이 된다

③ 만일 $L_{N+1} > L_N$이면 $L_{N+1} > L_N = N + 1$이 된다.

만일 주어진 수열이 실제 난수열(truely random sequence)인 경우에 선형 복잡도 프로파일은 N이 증가함에 따라 $n/2$ 선을 따라 불규칙하게 증가하게 되며 난수열의 선형 복잡도 프로파일은 주기에 거의 근접하게 된다.

17) Zip-Lempel 복잡도 검정법

ZLC(Ziv-Lempel Complexity)는 1976년 Ziv와 Lempel이 제안한 Complexity Measure로서 현재 주로 Data Compression 등에 많이 이용되어지고 있으며, 본 절에서는 ZLC를 이용한 이진 수열의 임의성 특성을 분석하여 스트림 암호의 임의성을 측정하는데 사용하고자 한다.

[정의] ZLC(Ziv-Lempel Complexity)

수열 ($s^n = s^1, s^2, \cdots, s^n$)을 길이 n의 이진 수열이라 두자. s_1다음에 slash를 삽입하고, i번째 slash가 $s_k, 1 \le k_i \le n-1$ 다음에 있다고 하면, $i+1$번째 slash는 $s_{k_{i+1}}$ 다음에 삽입된다. 여기서 $k_{i+1} = k_i + L_i + 1 \le n$이고 L_i는 이고 $s_{p_{i+L_i-1}} = s_{k_i+1}, \cdots, s_{k_i, L_i}$이고 $s_{k_i+1}, \cdots, s_{k_i, L_i}$에 대하여 $p_i \in Z(1 \le p_i \le k_i)$가 존재하는 부분 문자열의 최대 길이이다. 이때, 수열 (s^n)의 ZLC는 다음처럼 계산된다.

$$ZLC = \begin{cases} \#\ of\ \text{slash, if } s_n \text{ is followed by a slash} \\ \#\ of\ \text{slash} + 1, \text{otherwise} \end{cases}$$

ZLC 검정은 수열의 반복성의 정도를 나타내는 측도이며 수열에서 어떤 한 개의 부분 수열도 이전에는 일어나지 않도록 분해되어진 부분 수열의 개수, 즉 수열을 따라 움직일 때 나타나는 새로운 패턴의 개수를 말한다. ZLC는 아래에서 패턴의 길이를 구하는 알고리즘 A와 이를 이용하여 ZLC를 구하는 알고리즘 B에 의해 쉽게 계산될 수 있다.

<검정 알고리즘 A> 패턴의 길이를 구하는 알고리즘

단계 1 : 초기화

1.1 패턴 길이 l에 2를 대입한다.

1.2 n+1은 새로운 패턴이 나타나는 수열의 위치를 나타내도록 한다.

단계 2 : $\Im = \{j \in N : s_j = s_{n+1}, 1 \le j \le n\}$를 정의한다.

2.1 $\Im = \varnothing$ 인 동안 다음의 수행을 반복한다.

2.2 모든 $j \in \Im \ni s_{j+l=1} \neq s_{n+1}$를 삭제한다.

2.3 만일 $\Im = \varnothing$ 이면, $l < -l + 1$로 둔다.

2.4 만일 위 조건을 만족하는 j가 존재하지 않는다면, 패턴 길이 l을 반환하고 수행을 멈춘다.

<검정 알고리즘 B> ZLC

단계 1 : 초기화

1.1 slash에 1을 대입한다.

1.2 전체 수열의 길이를 2로 둔다.

단계 2 :

2.1 전체 수열의 길이가 입력 수열의 길이보다 작을 동안 다음의 수행을 반복한다.

2.2 slash값을 1 증가시킨다.

2.3 전체 수열의 길이에 대해 알고리즘 A에서 구한 패턴 길이를 계속 더해 나간다.

단계 3 : ZLC 값으로 slash를 반환하고 수행을 멈춘다.

ZLC의 특성은 다음과 같다.

① 주기 p인 수열에서 $2p-1$항 이후까지의 ZLC는 $2p-1$항까지의 ZLC와 같다.

② s_{n+1}에서 시작하는 패턴의 평균 길이는 $[\ \log_2 n\ ] + 1$ 이다.

위의 특성과 알고리즘 B를 이용하면, 주어진 이진 수열의 동일한 패턴의 반복정도를 알 수 있다. 즉, 두 번째 성질을 이용하면, 평균 패턴 길이와 ZLC를 구할 수 있고, 어느 이진 수열의 각 패턴 길이와 ZLC를 구하여 이와 비교한 후, 평균값에 크게 미치지 못하는 수열은 랜덤하지 않은 수열로 판정할 수 있다.

18) Walsh-Power Spectrum 검정법

서로 독립이고 일양 분포를 갖는(independent and identically distributed) 확률 변 수 열의 자기 상관 함수(autocorrelation function)를 δ-함수라 한다. 자기상관 함수가 δ-함수가 된다는 것은 그 수열의 Walsh Power Spectrum이 평평하다는 것과 동치이다. Walsh-Power Spectrum 검정은 이러한 사실을 이용하여 주어진 수열의 무상관성(uncorrelatedness)을 검정하는 통계적 검정법이다. Power Spectrum 검정은 자기상관 함수에 감춰져 있을지 모르는 어떤 주기성을 찾아서 관측 가능하도록 확대하는 것과 같은 효과를 지닌다.

Walsh 함수는 다음과 같이 반복적으로 정의된 직교 함수열로서 정의된다.

$$Wal(0,t) = 1$$

$$Wal(2i,t) = \begin{cases} Wal(2i+1,t) = Wal(i,2t), & \text{for } 0 \le t \le \frac{1}{2}, \\ -Wal(2i+1,T) = wAL(i,2t-1), & \text{for } \frac{1}{2} \le t \le 1, \\ i = 0,1,2,\ldots \end{cases}$$

Walsh 함수를 구하는 방법 중 하나는 Rademacher 함수의 곱으로부터 구하는 방법이 있는데 다음과 같다.

우선, Rademacher 함수 Rad(n, t), 0≤t≤1은 다음과 같이 정의한다.

$$Rad(n,t) = sign(\sin(2^n \pi t)), \quad n = 0,1,2,\cdots$$

Paley-ordered Walsh 함수 $Pal(n,t)$는 다음과 같이 정의한다.

$$Pal(n,t) = \prod_{i=0}^{m} Rad(i,t)^{b_i}$$

여기서, b_i들은 n을 이진수로 표현했을 때 i번째의 값이다.

$$n=(b_m b_{m-1}\cdots b_1)_2=\sum_{i=0}^{m} b_i 2^{i-1}.$$

예를 들어,
13 = $(1101)_2$이므로 $Pal(13,t), Pad(4,t), Rad(3.t), Rad(1,t)$이다.

$$T_i=\sum_{j=0}^{2^m-1}(X_{j+i2^m})^2, i=0,1,2,\cdots$$

power Spectrum의 근사값 계산을 위하여 band spectrum estimate를 다음과 같이 정의한다.

$$E(T_i)=2^m o^2, Var(T_i)=2^{m+1} o^4$$

T_i의 평균과 분산은 각 각 이다. X_i가 정규분포를 따르므로 T_i는 자유도 2^m인 x^2-분포를 따른다. 그러나 m이 충분히 크면 T_i는 근사적으로 정규분포를 따른다. 그러므로, Z_i는 중심극한정리에 의해 근사적으로 정규분포를 따른다.

$$Z_i=\frac{T_i-2^m\sigma^2}{2^{(m+1)/2}\sigma^2}, i=0,1,2,\cdots$$

Walsh Band Spectrum을 이용한 난수성 검정법에서는 주어진 수열 $\{v_n\}_{n=0}^{2^n-1}$의 Walsh Power Spectrum을 이용하여 무상관성을 검정한다. 이 경우 Walsh coefficient는 다음과 같다.

$$X_i=\sum_{j=0}^{2^n-1} Pal(i,j)u_j, \quad i=0,1,\cdots,n-1$$

z_a를 표준정규분포의 상위 100% 백분위 수, 즉, $\int_{-\infty}^{z_a}\frac{a}{\sqrt{2\pi}}e^{-\frac{a^2}{2}}dx=1-a$일 때 $N(a)$를 $z_{\frac{a}{2}}$보다 큰 $|Z_i|$의 개수라고 하자. 그러면 $N(a)$는 평균이 $a2^{n-m}$인 이항분포를 따르고 다음과 같이 정의된다.

$$N(a)=\#\{i\,|\,|Z_i|>z_{\frac{a}{2}}\}$$

2^{n-m}이 충분히 클 때 이항분포를 정규분포로 근사하여 $N(a)$가 아래와 같은 유의수준 β의 신뢰구간에 속하면 주어진 수열은 Walsh-Power Spectrum 검정을 통과하게 된다. 신뢰구간은 다음과 같다.

$$(a2^{n-m} - z_{\frac{\beta}{2}}\sqrt{a(1-a)2^{n-m}},\ a2^{n-m} + z_{\frac{\beta}{2}}\sqrt{a(1-a)2^{n-m}})$$

19) 새로운 일반적 검정(New Universal Test)

새로운 일반적 검정에서 구현된 기본 개념은 다음 비트 검정(Next Bit Test) 이론을 바탕으로 하였으며 수열에 관한 확률적 통계치가 주어진다면 이를 이용하여 추측할 수 있는 다음 비트들에 관한 정보를 얻을 수 있게 된다는 점을 이용하였다. 본 검정에서는 난수 발생기의 난수성 평가시 입력 수열에 대한 크기의 크고 작음에 관계없이 모든 임의 길이의 수열에 적용될 수 있도록 하였으며 이는 난수 발생기를 이용한 암호 시스템의 안전성 평가에 있어서 매우 유용하게 사용될 수 있을 것이다.

비트 생성기들에 대해 다음 비트 검정이 일반적인 검정으로서의 성질을 가진다는 것이 Yao에 의해 증명되었다. 다음 비트 검정은 비트 발생기에 의해 생성되어진 수열의 임의의 i 비트들에 대해 1/2 이상의 성공 확률을 가지고 i 비트 수열의 다음 비트를 예측하려는 것이다. 비트 발생기가 새로운 일반적 검정을 통과할 경우 비트 발생기들에 대한 어떠한 다른 검정도 통과한다는 관점에서 이 검정을 일반적이라고 평가할 수 있다.

Schrift와 Shamir에 의해 아래와 같은 기호와 정의들이 사용되어진다.

s_1^n : $\{0,1\}^n$상에서길이 n을 갖는 이진수열

s_i : 수열의 i번째 비트

s_1^k : j번째 비트에서부터 k번째 비트까지의 수열

$O(v(n))$: $v(n)$의 복잡도를 가지는 Big-Oh 표기

[정의] 수열 S_n이 $\{0,1\}^n$상에서의 확률 분포이면 앙상블 S는 수열 $\{S_n\}$된다.

[정의] S_n이 모든 n에 대해 균일(uniform)한 확률 분포이면 앙상블 S는 균일적이라 한다. 즉, 모든 α∈ $\{0,1\}^n$에 대해 $\Pr ob\{s_1^n = a\} = \frac{1}{2^n}$ 이다.

단, $\Pr ob(E)$: 확률 분포가 앙상블 입력 수열 S에 의해 정의되어 질 때 사건 E가 발생할 확률이다.

[정의] 다음 비트 검정(Next Bit Test)

만일 임의의 i와($1 < i \leq n$) 모든 확률적 다항식 시간 알고리즘 A : $\{0,1\}^n \rightarrow \{0, 1\}$, $|\Pr | \Pr ob_s\{s_1^{i-1} = s_i\} - \frac{1}{2} | \leq O(v(n))$이면 입력 수열 S는 다음 비트 검정을 통과한다고 정의한다.

Schrift와 Shamir는 편향된 입력 수열에 대해 "예측 또는 검정 통과(Predict or Pass Test)" 라는 검정을 소개했는데 여기에서 편향된 입력 수열은 다음과 같이 정의된다.

[정의] 모든 i에 대해 $\Pr ob_s\{s_i = 1\} = b$이라면 입력 수열 S는 고정된 편향 b를 지니고 1쪽으로 편향되어 진다 $\frac{1}{2} \leq b < 1$). 만일 모든 비트들이 독립적이라면 입력 수열은 독립된 편향이라 한다.

[정의] POP(Predict or Pass) 검정

만일 모든 $i(1 < i \leq n)$, 모든 고정 값 c, 모든 확률적 다항식 시간 알고리즘 A : $\{0,1\}^{i-1} \rightarrow \{0,1,?\}$에 대해서, $\Pr ob\{A(s_1^{i-1}) \neq ?\} \geq \frac{1}{n^c}$ 이면 $|\Pr ob_s\{A s_1^{i-1} = s_i \mid A(s_1^{i-1} \neq ?)\} - b| \leq O(v(n))$이고 편향된 입력 수열 S는 POP 검정을 통과한다고 정의한다.

[정리] 입력 수열이 완전히 독립적인 편향된 수열이 될 필요 충분 조건은 POP 검정을 통과하는 것이다.

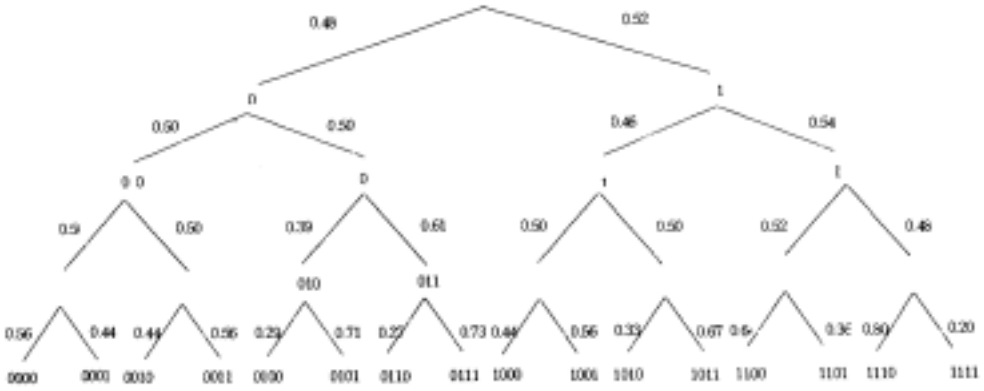
[정의] 임의 확률적 다항식 시간(구별) 알고리즘 D : $\{0,1\}^n \rightarrow \{0,1\}$에 대해서 $|\Pr ob(D(s_1)=1)-\Pr ob(D(s_2)=1)| \leq O(v(n))$이라면 두 개의 입력 수열 S_1과 S_2는 다항식적으로 구분 불가하다.

[정의] 편향 b를 가지는 독립적으로 편향된 입력 수열로부터 소스 S가 다항식적으로 구분 불가라면 수열 S는 고정 편향 b를 가지는 완전 독립 편향된 입력(perfect independence biased source)이 된다.

위에서 살펴 본 확률적 검정과 POP 검정이 서로 연관되어질 수 있다는 데에 바탕을 두고 실제적인 새로운 일반적 검정을 소개한다. 가중 트리를 고려하는데, 각 노드에는 각 계층상에서 발생하는 패턴의 발생 횟수를 적고 각 노드에서 바로 아래에 연결되는 간선들에는 상위 계층상에서 발생하는 패턴의 수에 대한 하위 계층에서 발생하는 패턴의 수의 확률 비를 적어내려 가면서 트리를 구성해간다. 계산상의 복잡성을 피하기 위해 계층 l에서의 패턴 비교를 위해 수열의 처음 $l-1$ 비트를 수열의 끝 부분에 덧붙여 준다. 상위 계층에서 각 패턴들의 수는 그 아래 두 노드들 간에 직접적으로 관련된 패턴들의 수의 합으로부터 계산되어질 수 있다. 다음은 트리의 구성 예이다.

예제 다음과 같은 이진 수열이 주어졌을 경우 본 검정에서 제안한 방식으로 비트의 발생 패턴과 발생 횟수에 따른 확률 비를 적어내려 가면서 트리를 구성해 본다.(n=75)

010101110010111001111011100110000100100111000110101110000010111101
100000111



트리에서 아래로 이동함에 따라 확률들이 1/2에서 0과 1로 편차가 생기게 됨을 알 수 있으며 이것은 임의의 수열에 대해서 일반적인 성질이다. 즉, 트리에서 더 낮은 계층으로 내려감에 따라 패턴들의 발생 횟수는 점점 더 적어지게 된다. 이러한 성질은 포커 검정처럼 큰 m에 대해서는 통계적 검정의 수행 결과를 무의미하게 만든다. 하지만, 이것은 더 낮은 계층에서 통계량이 주어진다면 큰 m을 가지는 많은 패턴들의 다음 비트를 예측할 수 있음을 알려준다. 예를 들어, 패턴 1010이 예제1의 수열에서 나타난다면 1의 확률로 다음 비트는 1이 됨을 알 수 있다.

75-비트 수열에 대해서 만일 편차 값이 $0.3869 \le b \le 0.6131$라면 수열은 빈도 검정을 통과하게 되고 그렇지 않다면 의사 난수열이 되는 것이 기각될 것이다. 이러한 검정은 다른 길이를 가지는 임의의 다른 수열에 대해서도 확장할 수도 있다.

[정의] s_1^n은 길이 n을 가지는 수열이다. 다음 비트의 결정 임계값(threshold decision) α는 다음과 같이 정의된다.

$$\alpha = \frac{1+\sqrt{\frac{x^2}{n}}}{2}$$

단, x^2는 검정에서 요구되는 유의수준에 대응되는 값이다.

De bruijn 수열과 같이 높은 복잡도를 가지는 난수열들에 대해서 $n = 2^l$일 때, $l = \log_2(n)$인 계층에서 각 패턴은 한번씩 발생한다. 하지만 좀더 적은 복잡도를 지니는 수열들에 대해서는 몇몇 패턴들은 계층 l에서 발생하지 않고 나머지 패턴들은 한 번이나 한 번 이상 발생한다. 이것은 계층 l-1의 노드에서 계층 l까지의 노드에 놓여있는 간선들 몇몇은 확률이 1(또는 0)이 되게 된다. 또한 이러한 확률들이 특정 패턴의 발생과 관련되어 있음을 고려한다면 수열의 일부분을 재구성할 수도 있다.

재구성에 관한 임계값(threshold)을 정의하기 위해 POP 검정의 확장인 아래와 같은 정의와 정리를 고려해 보자.

[정리] Extended POP 검정

각 $i,l(1 < i,l \le n)$, 고정된 상수 c, 모든 확률적 다항식 시간 알고리즘 A : $\{0,1\}^{i-1} \rightarrow \{0,1^l,?\}$에 대해 편향된 소스 S가 만약 $\Pr ob\{A(S_1^{i-1}) \neq ?\} \geq \frac{1}{n^c}$ 일 때, $|\Pr ob\{A s_1^{i-1} = s'^{i-1+l}_i \mid A(s_1^{i-1} \neq ?)\} - b| \leq O(v(n))$이 성립한다면, 확장된 POP 검정을 통과한다.

[정리] 다음의 조건들은 동치이다.

① 편향된 소스는 완전 독립적 편향된 소스이다.

② 편향된 소스는 확장된 POP 검정을 통과한다.

만일 소스가 확장된 POP 검정을 통과할 수 없다면 주어진 부분수열 블록이 다음 블록을 효율적으로 추측할 수 있는 확률적 다항식 시간 검정이 존재함을 의미한다. 따라서, 검정 A는 주어진 이전블록으로부터 s_i를 예측할 수 있고 소스는 완전 독립적 편향된 소스가 아니다.

따라서 빈도 검정을 통과하는 의사 난수열에 대해 부분 수열의 블록 당 다음 비트의 확률은 계산된 편향 값을 넘지 않아야 한다. 이것은 필요조건이므로 계산된 편향 값이 확률적 다항식 시간 알고리즘 A에 대한 결정 임계값이 되도록 정의할 수 있다. 수열에서의 다음 비트가 편향치 보다 더 높은 확률을 가지고 나타날 때, 즉 $b \le p$ or $p \le 1-b$, 알고리즘은 수열에서 그 다음 비트를 예측할 수 있다. 수열에서의 다음 비트가 편향치보다 적은 확률을 가지고 나타날 때, 즉 $1-b \le p \le b$일 때 알고리즘은 다음 비트를 예측하지 못하게 된다.

다음은 새로운 일반적 검정의 알고리즘이다.

<검정 알고리즘>

길이 n을 갖는 수열에 대한 새로운 검정 알고리즘

단계 1 : 결정 임계값 α 값을 다음과 같이 계산한다..

$$\alpha = \frac{1+\sqrt{\frac{\chi^2}{n}}}{2}$$

단계 2 : $l = round(\log_2(n))$을 계산한다.

단계 3 : 수열의 꼬리에 수열의 처음 부분에 나타나는 $l-1$개의 비트들을 덧붙이고 수열을 서로 겹쳐 가면서 l 비트의 단위로 나눈다.

단계 4 : 각각의 블록을 비교해 나가면서 길이 l 을 갖는 각 패턴의 발생 횟수를 계산한다.

단계 5 : 계층 l 과 $l-1$에서 트리를 형성해 나가면서 각 간선에 대응되는 확률을 구한다.

단계 6 : 계층 $l-1$에 있는 각 노드에 대해 만일 다음 비트가 α보다 더 높은 확률을 가지고 나타난다면 다음 비트는 예측되어질 수 있으며 그렇지 않은 경우 다음 비트는 결정될 수 없다.

단계 7 : 계층 *l*-1에 있는 각 노드에 대해 이후에 예측되어질 수 있는 수열의 길이를 계산한다.
위의 알고리즘을 사용하여 수열의 국소적 비난수 성향과 전역적 비난수 성향을 다음의 방법으로 평가할 수 있다.

① 국소적 비난수(Local non-random) 성향

만약 계층 *l* - 1에서 *l* + 1보다 많은 비트들이 예측되어 질 수 있는 임의의 노드가 존재한다면 다음 블록이 예측될 수 있는 길이 *l* 의 블록이 존재함을 의미한다. 따라서, 수열에서 국소적 비난수성이 존재하고, 수열은 요구되는 성질을 만족하는 생성기로는 기각되어질 수도 있다. 위의 검정은 상위 계층의 가지들에 나타나는 확률이 1/2로부터의 편차가 더 낮은 가지들에 더 많은 편차를 유발시킬 것이라는 관점에서 만능 검정의 의미가 있다. 즉, 새로운 검정에서 편차는 수열에 가중되어지는 local non-random behavior로서 나타내어질 것이며 반면에 기존의 통계적 검정의 각 종류는 트리의 상위 계층 가지들에서 1/2로부터의 편차의 측정치에 대한 것이다. 따라서 수열이 새로운 검정을 통과한다면, 기존의 빈도, 계열 및 포커 검정과 같은 표준적인 통계적 검정을 통과하게 된다.

② 전역적 비난수(overall non-random) 성향

수열의 전반적인 성향에 대한 평가를 위해 새로운 검정의 결과를 이용할 수 있다. 이것은 다음에 예측되어지는 비트들의 수에 대한 노드들의 수가 주어지는 히스토그램을 형성함으로써 가능해 진다.
난수 수열에 대해, 계층 *l* - 1과 계층 *l* 사이의 가지들에서 나타나는 확률은 (0, 1)사이의 값을 가지는 확률 변수가 된다.

이러한 확률이 (1-α,α)의 범위를 초과한다면 다음 비트를 예측할 수 있게 된다. 그러나, 만일 확률이 위의 범위 내에 존재한다면 다음 비트를 결정하거나 예측할 수 없다. 이러한 (2α-1)의 범위를 β 라 부르기로 하자. 각 노드에 대해, 다음 비트가 예측되어진다면 만족하는 노드들의 경로를 그려보자. 예를 들어, 추측되는 다음 비트가 1 인 노드 00101로부터 노드 01011를 추측하게 된다. 이러한 단계의 노드들을 단계-1 노드라 부른다. 다시 노드 01011에 대한 다음 노드는 10111이 되고 이러한 단계에서의 노드들을 단계-2라 부른다. 이러한 방법으로 서로. 다른 단계에 있는 노드들의 수를 계산할 수 있게 된다.

De bruijn 수열과 같은 복잡한 수열에 대해, 단계-0 노드의 수는 $N_0 = 2^{round(\log_2(n)-1}$ 이 된다. 많은 노드들에 대해서, 모두 다음 비트를 예측할 수는 없는데, 그러한 노드들의 수는 $\aleph_0 = BN_0$와 동치이다. 다음 비트를 예측할 수 있는 노드들에 대해서는 다음 비트를 추측하고 한 단계씩 더 확장할 수 있다. 이러한 노드들의 수는(1-β) N_0개의 노드를 가진다. 다음 비트를 예측할 수 있는 노드만이 stage-1 단계에 도달하는 것 은 아니다. 단계-1 노드들의 수는 $N_1 = N_0(1-B)(1-\frac{r}{2})$이 되고, γ 는 두 노드가 다음 단계의 같은 노드에 도달하는 확률이다. 예제 1의 수열에서 노드 01011과 노드 11011의 뒤에 비트 1이 추측되어 질 수 있다. 따라서 양쪽 노드들은 모두 노드 10111에 도달한다. 단계-1에 있는 많은 노드들로부터 계속 확장할 수는 없는데, 이와 같은 노드들의 수는 $\aleph_1 = B N_1$ 이다. 단계-1 노드들에 도달하는 단계-0 노드들의 수는 $NA_1 = \frac{\aleph_1}{1-\frac{r}{2}}$ 이 된다. 즉, NA_1는 이후의 한 비트가 예측되어질 수 있는 계층 l-1에 있는 노드들의 수이다.

단계-2 노드들의 수는 $N_2 = N_1(1-\beta)(1-\frac{\chi}{2})$가 된다. 한 단계 더 확장할 수 없는 단계-2 노드의 수는 $\aleph_2 = B N_2$가 되며, 단계-0 노드들에 대응하는 노드들의 수는 $NA_2 = \frac{\aleph_2}{(1-\frac{r}{2})^2}$가 된다. 일반적으로, 단계-i 노드들의 수 $N_i = N_{i-1}(1-B)(1-\frac{r}{2})$가 된다. 더 이상 확장할 수 없는 단계-i 노드들의 수는 $\aleph_i = B N_i$ 이고 단계-0에 대응하는 노드들의 수는 $NA_i = \frac{\aleph_i}{(1-\frac{r}{2})^i}$가 된다. 다시 한번, 이후의 비트들이 예측되어 질 수 있는 계층 l-1 노드들의 수는 NA_i가 됨을 유의하자.

위의 계산에 대해, r 에 대한 하한 경계는 다음과 같이α로부터 계산되어질 수 있다. 독립적인 소스로 이루어지는 트리에서, 만일 두 노드의 처음 부분이 서로 다르고 꼬리 부분이 동일하다면, m-2 비트 패턴을 가진 두 노드들은 다음 단계에서 합병될 것이다. 특정한 m-1 비트 패턴이 주어졌을 때 두 노드들의 꼬리 부분이 동일한 확률로서 r를 계산할 수 있다. 모든 노드들이 레이어 l -1에서 동일한 확률을 가지고 모든 단계들에서 충돌이 발생할 확률이 동일하다고 가정하면 $r = 1 + 2a_2 - 2a$가 된다. 단계-0 노드들의 수는 2^{l-1}과 동일하고 l +1이 되는 다음 단계의 수를 정의한다면 결과값들을 평가하기 위한 히스토그램을 그릴 수 있다.

사. 부가정보 이용 공격

1) DPA(Differential Power Analysis)

DPA 공격은 스마트카드와 보안 암호토큰에 대한 공격방법으로써 트랜지스터 논리 게이트와 스마트카드에서 동작하는 S/W 또는 다른 암호장치 등에서의 특이 행동을 탐색하는 것이다.

일반적으로 DPA 공격은 장치의 전자적 행동을 모니터한 후, 장치 안의 비밀 정보(비밀키 또는 사용자 PIN 등)을 결정하기 위하여 고도의 통계적 방법을 사용하여 수행된다.

http://www.cryptography.com/dpa/index.html

2) DFA(Differential Fault Analysis)

본 공격법은 Biham과 Shamir에 의해서 제안된 것으로서 스마트카드와 같은 특정 하드웨어 형태의 tamperproof된 암호시스템에 마이크로 웨이브 혹은 이온을 쬐어 암복호화 과정 중 어떤 하나의 동작에서의 레지스터를 비정상적으로 동작시킴으로서 한 비트에서 발생되는 오류의 확률을 정보로 하여 공격하는 방법이다. 공격자는 라운드 및 비트 위치를 모르고 공격하는 것이다. 공격자가 tamperproof된 암호시스템을 가지고 외부에서 물리적인 충격만으로 같은 평문과 키를 반복적으로 입력할 수 있다라는 가정이 있어야 한다. 그러한 동작을 반복한 결과 하나의 평문과 키로서 생성된 서로 다른 두 개의 암호문이 발생되는데 하나는 정확하게 발생된 암호문이고 다른 하나는 오류가 발생된 암호문이다. 이론적으로 DES, 3중DES(168비트 키) 그리고 768비트의 서브키로서 DES를 분석한다. 이러한 분석을 일반화시켜 라운드 함수 및 키 스케줄 등에도 적용시킬 수 있다. Differential Fault Analysis는Skipjack과 같은 공개되지 않았던 암호알고리즘을 분석하는데 초점을 맞춘 것이다.

3) Timing 공격

C. Kocher에 의해 CRYPTO'96[18]에서 제안된 공격인 Timing attack은 암호화 처리시간이 입력형태에 따라 약간의 차이가 있다라는 점에서 착안한 공격이다. 불필요한 연산들, branching 및 조건문, RAM cache hits, 덧셈 및 곱셈 등과 같은 연산들은 데이터에 의해서 최적화하기 때문에 암호처리 속도는 암호화 키와 입력 평문에 의존함을 알 수 있다. 공개키 암호알고리즘에서도 비밀키 연산속도를 측정하여 보면 공격자는 고정된 길이의 Diffie-Hellman 지수승이나 RSA키의 인수(소인수 분해한 하나의 인수)를 찾을 수 있는 단서를 구할 수 있을 것이다. 또한 대칭키 암호알고리즘에도 적용 가능하다. 예들 들면 56비트 DES키를 각 28비트 길의 C, D로 분할하여 키 스케줄에 의하여 한 비트 씩 회전하도록 조건을 줄 수 있다.

입력 키에 영인 비트가 많으면 많을수록 key setup 시간이 오래 걸릴 수 있다. DES 암호화 처리 속도로 인하여 키의 hamming weight에 정보를 평균적으로 $\sum_{n=0}^{56} \frac{\binom{56}{n}}{2^{56}} \log_2 \left\lfloor \frac{\binom{56}{n}}{2^{56}} \right\rfloor \approx 3.95$ 비트 노출시킬 수 있다. IDEA는 모듈러 $2^{16}+1$에서의 곱셈을 사용하므로 모든 데이터들을 가지고 고정된 시간에 곱셈연산을 수행할 수 없으며 RC5에서는 가변적인 회전연산을 사용하므로 모든 cpu등 platform에서 고정된 시간에 회전연산을 수행할 수 없다. RAM cache 성능에 따라 메모리에 저장되는 table을 사용하는 암호알고리즘의 암호화 속도에도 차이가 있다. 현재까지는 특정 시스템에서의 특정 암호알고리즘에 대한 timing attack공격은 이론적으로만 연구되어 왔다.

---

18) "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and another systems", Advances in Cryptology: Proc. of CRYPTO'96, Springer-Verlag, Berlin, pp 104-133, 1996

제 5 장 결 론

암호알고리즘의 평가는 평가의 신뢰성이 주요 이슈일 것이다. 그러나 실제 암호알고리즘을 평가하는 일은 매우 난해한 일이며 특히, 암호알고리즘의 안전성 평가는 많은 어려운 문제를 안고 있다. 현재 알려진 여러 공격에 대한 안전도를 이론적으로 평가할 수 있는 경우도 있지만, 그렇다고 해서 미지의 공격이 없다고 단언하기는 곤란한 일이기 때문이다. 즉, 암호알고리즘의 안전성 평가는 설계가 그렇듯 많은 경험과 시행착오에 의존하는 반복적이고 지루한 작업일 것이다.

본 고에서는 기존에 알려진 암호알고리즘 안전성 평가항목을 분석 · 정리하였다. 본 고에 나열된 평가항목을 다 만족하였다고 하여 그 암호알고리즘이 안전한 것은 아니며, 다만 본 평가 항목을 통과한 암호알고리즘은 현재까지의 공격에 안전하다는 것만을 의미한다. 따라서 본고에서 제시된 각 항목은 안전도를 검증하기 위한 최소 통과 기준이며, 암호알고리즘의 분석 및 평가의 기본 자료로 활용 가능하다. 향후, 평가를 수행하기 위한 분석 조건과 자동화 도구 개발가능성 등 다양한 측면에서의 재분석이 필요하다고 하겠다.

[첨부1] 주요 旣 개발 블록 암호알고리즘 명세

http://www.users.zetnet.co.uk/hopwood/crypto/guide/scan/index.html

■ 3-Way

o 설계자 : Joan Daemen
o 공개 시기 : 1994
o 키길이 : 96 비트
o 블록 크기 : 12 바이트
o 참고문헌:

[정의, 분석l Joan Daemen, "Cipher and Hash Function Design, Strategies based on linear and differential cryptanalysis" Ph.D. Thesis, Katholieke Universiteit Leuven, March 1995. http://www.esat.kuleuven.ac.be/~cosicart/ps/JK-9500/(in particular see chapter 7, "block cipher design")

[정의, 분석] J. Daemen, R. Govaerts, J. Vandewalle, "A New Approach to Block Cipher Design," Fast Software Encryption, Cambridge Security Workshop Proceedings, Volume 809 of Lecture Notes in Computer Science (R. Anderson, ed.), pp. 18-32. Springer-Verlag, 1994.

[관련정보] Bruce Schneier, "Section 14.5 3-Way," Applied Cryptography, Second Edition, John Wiley & Sons, 1996.

[분석] J. Kelsey, B. Schneier, D. Wagner "Key-Schedule Cryptanalysis of 3-WAY, IDEA, G-DES, RC4, SAFER, and Triple-DES", Advances in Cryptology -Crypto '96 Proceedings, pp. 237-251. Springer-Verlag, August 1996.
http://www.counterpane.com/key_schedule.html

[분석] J. Kelsey, B. Schneier, D. Wagner, "Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA", ICICS '97 Proceedings, Springer-Verlag, November 1997.
http://www.counterpane.com/related-key_cryptanalysis.html
[Test] wei Dai, Crypto++3.0, file 3wayval.dat
http://www. eskimo.com/~weidai/cryptlib.html
o 안전도관련 코멘트:
- 3-Way는 연관키 공격에 취약
(그러므로 강한 PRNG로 키를 생성하여 사용하거나, 비트의 소스가 <해쉬함수의 출력과 같이> 충분히 상관성이 없어야 함)

■ AES

o 설명
- NIST의 AES(Advanced Encryption Standard) 콘테스트 승자에게 붙여질 이름
· 별명 : "AES128", "AES192", "AES256", OpenPGP.Cipher.7", "OpenPGP.Cipher.8", "OpenPGP.Cipher.9"
o 참고문헌 :
[정의] NIST, AES Home Page,
http://www.nist.gov/aes/
[정의] AES Round 1 - Documentation Download,
http://www-08.nist.gov/encryption/aes/roundl/docs.htrn
[정보] CAESAR - AES 후보 알고리즘 분석 및 검토를 위한 프로젝트
http://www.dice.ucl.ac.be/crypto/CAESAR/caesar.html
[정보] Lars Knudsen, Vincent Rijmen, The Block Cipher Lounge - AES, http://www.ii.uib.no/~larsr/aes.html
[정보] John Savard, Towards the 128-비트 Era - AES Candidates,

http://fn2.freenet.edmonton.ab.ca/~jsavard/co0408.html
[분석] Eli Biham, "A Note on Comparing the AES Candidates," Submitted to the 2nd AES Conference.
http://csrc.nist.gov/encryption/aes/round1/conf2/papers/biham2.pdf
[분석] Olivier Baudron, Henri Gilbert, Louis Granboulan, Helena Handschuh, antoine Joux, Phong Nguyen, Fabrice Noilhan, David Pointcheval, Thomas Pornin, Guillaume Poupard, Jacques Stem, Serge Vaudenay, “Report on the AES Candidates," Submitted to the 2nd AES Conference.
http://csrc.nist.gov/encryption/aes/round1/conf2/papers/baudron1.pdf
[분석] G. Carter, E. Dawson, L. Nielsen, "Key Schedule Classification of the AES Candidates," Submitted to the 2nd AES Conference.
http://csrc.nist.gov/encryption/aes/round1/conf2/papers/carter.pdf
o 키 길이 : 128, 192 또는 256 비트(다른 키 길이도 지원 가능)
o 블록 크기 : 16 바이트(다른 블록크기도 지원 가능)
o 특허 현황 :
- 로얄티 및 특허료 없이 전 세계적으로 활용 가능함

■ Blowfish
o 설계자 : Bruce Schneier
o 공개 시기 : 1994
o 별명 : “OpenPGP.Cipher.4”
o 참고문헌:
[정의] Bruce Schneier, "Discription of a New Variable-Length Key, 64-bits Cipher (Blowfish)." Fast Software Encryption, Cambridge Security Workshop Proceedings, pp. 191-204 .Springer-Verlag, 1994.
http://www.counterpane.com/bfsverlag.html

[정보, 구현] Bruce Schneier, The Blowfish Encryption Algorithm page, http://www.counterpane.com/blowfish.html
[정보] Bruce Schneier, “Blowfish -- One Year Later," Dr. Dobb's Journal, September 1995.
http://www.counterpane.com/bfdobsoyl.html
[정보] Bruce Schneier, “Section 14.3 Blowfish," Applied Cryptography, Second Edition, John Wiley & Sons,1996.(Note: 부록의 C 소스코드에 버그가 있음 그러나 Eric Young의 C 참조 구현에는 버그가 없음)
[분석] S. Vaudenay, "On the weak keys of Blowfish," Fast Software Encryption, Third International Workshop, Volume l008 of Lecture Notes in Computer Science (B. Preneel, ed.), pp. 286-297. Springer-Verlag, 1995.
[테스트] Eric Young, Blowfish test vectors,
http://www.counterpane.com/vectors.txt
o 키 길이 : 최소 32, 최대 448, 8비트의 배수 ; 디폴트 128 비트
o 블록 크기 : 8 바이트
o안전도관련 코멘트 :
- S. Vaudenay 논문에서 기술한 취약키는 전체 16라운드 Blowfish의 중요 결점으로 나타나지는 않음

■ Cast-128
c 설계자 : Carlisle Adams, Stafford Tavares
o 공개 시기:1997
o 공개 시기 : 1997
o 별명 : “CAST5," ”Open PGP. Cipher. 3"
o 참고문헌 :
[정의, 테스트] Carlisle Adams, "The CAST-128 Encryption Algorithm," RFC 2144. May 1997.
[정보, 분석] CAST Encryption Algorithm Related Publications,

http://adonis.ee.queensu.ca:8000/cast/
[정보] Carlisle Adams, "Constructing Symmetric Ciphers Using the CAST Design Procedure," Selected Areas in Cryptography (E. Kranakis and P. van Oorschot, ed.), pp. 71-104. Kluwer Academic Publishers, 1997, and Designs, Codes, and Cryptography, Vol. 12, No. 3, pp. 283-316, 1997.
http://www.entrust.com/downloads/cast.ps(HTML version)
Also "CAST Design Procedure Addendum,"
http://www.entrust.com/downloads/castadd.ps
o 키 길이 : 최소 40, 최대 128, 8비트의 배수; 디폴트 128 비트
o 블록 크기 : 8 바이트

■ Cast-256
o 설계자 : Carlisle Adams, Howard Heys, Stafford Tavares, Michael Wiener
o 공개 시기 : 1998.6.
o 참고문헌 :
[정의, 분석] Carlisle Adams, The CAST-256 Encryption Algorithm,
http:/www.entrust.com/resources/pdf/cast-256.pdf
[정보] Carlisle Adams, "Constructing Symmetric Ciphers Using the CAST Design Procedure," Selected Areas in Cryptography (E. Kranakis and P. van Oorschot, ed.), pp. 71-104. Kluwer Academic Publishers,1997, and Designs, Codes, and Cryptography, Vol. 12, No. 3, pp. 283-316, 1997.
http:/www.entrust.com/downloads/cast.ps(HTML version)Also "CAST Design Procedure Addendum,"
http://www.entrust.com/downloads/castadd.ps
(CAST-256을 기술하지 않았지만, 설계와 관련 있음)

[특허] Carlisle Adams, [[need patent title]]
U.S. Patent #5,511,123.[[need date]] Canadian Patent Application 2,134,410. Japanese Patent Application 6-295746. U.S. Patent Application 08/761,763. Canadian Patent Application 2,164,768. PCT Patent Application CA96/00782. U.S. Patent Application 08/895,875.
[테스트] NIST, CAST-256 Test Values,
http://www-08nist.gov/encryption/aes/round1/testvals/cast-256-vals.zip
o 키 길이 : 최소 128, 최대 256, 32 비트 배수; 디폴트 128 비트
o 블록 크기 : 16 바이트
o 특허 현황 :
Entrust Technologies, Inc.가 특허권 소유 (참고문헌). (로얄티는 없음)

■ CRYPTON-0.5
o 설계자 : 임채훈
o 공개시기 : 1998
o 설명 :
- AES 후보알고리즘으로 NIST에 제출된 버전임
o 참고문헌 :
[정의, 분석] Chae Hoon Lim, Hyo Sun Hwang, CRYPTON: A New 128-비트 Block Cipher - Specification and Analysis (Version 0.5),
http://crypt.future.co.kr/~chlim/pub/cryptonv05.ps.
[정보, 테스트] The CRYPTON: A new 128-비트 block cipher page.
http://crypt.future.co.kr/~chlim/crypton.html
[테스트] NIST, CRYPTON v0.5 Test Values,

http://www-08.nist.gov/encrytion/aes/round1/testvals/crypton-vals.zip
o 코멘트 :
- "CRYPTON: A New 128-bits Block Cipher - Specification and Analysis"는 바이트 치환(phi) 설명 오류가 있음.
· 그림 4의 오른 쪽 b33이 b03임
o 키 길이 : 최소 64, 최대 256,32 비트의 배수 ; 디폴트 128 비트
o 블록 크기 : 16 바이트

■ CRYPTON
o 설계자 : 임채훈
o 공개 시기 : 1998. 12.
o 별명 : “CRYPTON-1.0”
o 설명 :
- CRYPTON의 1.0 버전 (현재 최종 버전)
o 참고문헌 :
[정의, 분석] Chae Hoon Lim, Hyo Sun Hwang, CRYPTON: A New 128-비트 Block Cipher - Specification and Analysis (Version l.0),
http://crypt.hture.co.kr/~chlim/pub/cryptonv1O.ps.
[정보, 테스트] The CRYPTON: A new 128-비트 block cipher page http://crypt.future.co.kr/~chlim/crypton.html
o 키 길이 : 최소 0, 최대 256,8 비트의 배수; 디폴트 128 비트 (Note : CRYPTON-0.5와 다름)
o 블록 크기 : 16 바이트

■ CS-Cipher

o 설계자 : Jacques Stern Serge Vaudenay

o 공개 시기 : 1998

o 참고문헌:

[정의, 분석, 테스트, 구현] Serge Vaudenay, Jacques Stem, “CS-Cipher,” In Fast Software Encryption, Paris, France. Lecture Notes in Computer Science No. 1372, pp. 189-205, Springer-Verlag, 1998. ftp://ftp.ens.fr/pub/dmi/users/vaudenay/SV98.ps

[정보] CS-Cipher home page,
http://www.cie-signaux.fr/security/index.htm

[분석] Serge Vaudenay, "On the Security of CS-Cipher," In Fast Software Encryption, Rome, Italy, To appear in Lecture Notes in Computer Science, Springer-Verlag,1999.

o 키 길이 : 최소 0, 최대 128,8 비트의 배수; 디폴트 128 비트

o 블록 크기 : 8 바이트

o 특허 현황 :

- Compagnie des Signaux이 특허권 소유

■ DEAL

o 설계자 : Lars Knudsen

o 공개 시기:1998.5.

o 참고문헌:

[정의, 분석] Lars Knudsen, DEAL: A 128-비트 Block Cipher, February 1998 (revised May 15, 1998).

http://www.ii.uib.no/~1arsr/newblock.html(Note: 본 논문에 오류 있음. 아래 참조)
[분석] Stefan Lucks, On the Security of the l28-비트 Block cipher DEAL.
http://th.informatik.uni-mannheim.de/m/lucks/papers.html
[테스트] NIST, DEAL Test Values,
http://www-08.nist.gov/encrytion/aes/round1/testvals/deal-vals.zip
o 키 길이 : 128,192 또는 256 비트; 디폴트 128 비트
o 블록 크기 : 16 바이트
o 코멘트 :
- "DEAL: A 128-비트 Block Cipher"은 키 생성알고리즘 설명 부분에 오류 있음
· <4>는 <3?으로(3번 발생),<8>은 <4>로(2번 발생)
· 입력키와 Xor되는 상수값
0x8000000000000000, 0x4000000000000000, 0x2000000000000000와 0x1000000000000000.
o 안전도관련 코멘트 :
- "On the Security of the 128-비트 Block Cipher DEAL,"은 키길이가 192비트 경우 DEAL의 인가된 취약성을 기술
· 실용적인 공격은 아님
- Counterpane Systems 사의 John Kelsey는 연관 키 공격과 동치키를 찾음 has found some related-key attacks and equivalent keys for DEAL
· unpublished, NIST의 AES 포럼 사이트에 발표(DEAL이 암호 알고리즘으로 사용될 경우는 실용적이지 않으나, Davies-Meyer와 같은 해쉬함수 구축에 사용될 경우 문제 있음)

■ DES

o 설계자: Don Coppersmith, Horst Feistel, Walt Tuchmann, U.S. National Security Agency

o 공개 시기 : 1976

o 참고문헌:

[정의] U.S. National Institute of Standards and Technology NIST FIPS PUB 46-2 (supercedes FIPS PUB 46-1), "Data Encryption Standard", U.S. Department of Commerce, December 1993.

http://www.itl.nist.gov/div897/pubs/fip46-2.htm

[정보] U.S. National Institute of Standards and Technology, NIST FIPS PUB 74, "Guidelines for Implementing and Using the NBS Data Encryption Standard"

[정보] Bruce Schneier, “Chapter 12 Data Encryption Standard,” Applied Cryptography, Second Edition, John Wiley & Sons, 1996.

[정보] A. Menezes, P.C. van Oorschot, S.A. Vanstone, “Section 7.4 DES," Handbook of Applied Cryptography, CRC Press, 1997.

[분석] E. Biham, A. Shamir, "Differential Cryptanalysis of the Full 16-Round DES," CS 708, Proceedings of Crypto '92, Volume 740 of Lecture Notes in Computer Science, December 1991.

http://www,cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/1991/CS/CS0798.ps

[분석] E. Biham, A. Shamir, Differential Cryptanalysis of the Data Encryption Standard, Springer-Verlag, 1993.

[분석] M. Matsui, "Linear cryptanalysis method for DES cipher," Advances in Cryptology - EUROCRYPT '93 Proceedings, Volume 765 of Lecture Notes in Computer Science (T. Helleseth, ed.), pp. 386-397. Springer-Verlag, 1994.

[분석] E. Biham, A. Biryukov, "An Improvement of Davies' Attack on DES," CS 817, EUROCRYPT '94 Proceedings(May 1994), Volume 950 of Lecture Notes in Computer Science (A. De Santis, ed.), Springer Verlag, 1995, and Joumal of Cryptology, Vol. 10, No. 3, pp. 195-206, 1997.
http://www,cstechnion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/1994/CS/CS0817.ps
[분석] Lars Knudsen, "New potentially weak keys for DES and LOKI," Advances in Cryptology - EUROCRYPT '94 Proceedings, Volume 950 of Lecture Notes in Computer Science (A. De Santis, ed.), pp. 419-424. Springer Verlag, 1995.
ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/knudsen/potential.ps.Z
o 키 길이 : 코드화에 64비트(패러티 비트 배재시 56비트)
o 블록 크기 : 8 바이트
o 코멘트 :
- 구현시 키의 패러티 비트는 무시하여야 함(즉, 체크하지 말아야 함)
- 그러나, DIS의 KeyGenerators을 바른 패러티를 갖는 키를 출력하여야 함
o 안전도관련 코멘트 :
고정된 56-비트 키 길이는 키 검색기술을 막기에는 너무 짧음

■ DESede
o 설계자 : Whitfield Diffie, Martin Hellman, Walt Tuchmann
o 공개 시기 : 1978
o 별명 : "TripleDES", "DES-EDE3", "3DES", "'OpenPGP.Cipher.2"
o 참고문헌 :
[정보] U.S. National Institute of Standards and Technology, NIST FIPS PUB 46-2 (supercedes FIPS PUB 46-1),“Date Encryption Standard", U.S. Department of Commerce, December 1993.
http://www.itl.nist.gov/div897/pubs/fip46-2.htm

[정보] Bruce Schneier, “Chapter 12 Data Encryption Standard," and "Section 15.2 Triple Encryption," Applied Cryptography, Second Edition, John Wiley & Sons,1996.
[정의, 분석] R.C. Merkle, M. Hellman, "On the Security of Multiple Encryption," Communications of the ACM, vol. 24 no.7, 1981, pp. 465-467.
[분석] Paul van Oorshot, Michael Wiener, "A Known-Plaintext Attack on Two-Key Triple Encryption," Advances in Cryptology - EUROCRYPT '90 Proceedings, Volume 473 of Lecture Notes in Computer Science (I.B. Dmng?d, ed.), pp. 318-325. Springer-Verlag, 1991.
[정보, 분석] R.C. Merkle, Secrecy, authentication, and public key systems, UMI Research Press, Ann Arbor, Michigan, 1979.
[분석] J. Kelsey, B. Schneier, D. Wagner, "Key-Schedule Cryptanlysis of 3-WAY, IDEA, G-DES, RC4, SAFER, and Triple-DES", Advances in Cryptology - Crypto '96 Proceedings, pp. 237-251. Springer-Verlag, August 1996.
http://www.counterpane.com/key_schedule.html
[분석] Stefan Lucks, "Attacking Triple Encryption," Fast Software Encryption '98, Volume 1372 of Lecture Notes in Computer Science (S. Vaudenay, ed.), Springer-Verlag, 1998.
http://th.informatik.uni-mannheim.de/m/lucks/papers.html
o 키 길이 : 128 또는 192 비트; 디폴트 192 비트(코드화시) 112 또는 168 비트(패러티 배재시)
o 블록 크기 : 8 바이트
o 코멘트 :
- 패러티를 포함한 키 길이가 128 비트이면 (즉, 2키 3중 DES), 코드화 하는 첫 8바이트는 두 개의 외부 DES 연산을 위해 사용되는 키를 표현하고, 두 번째 8 바이트는 내부 DES 연산을 위해 사용되는 키를 표현한다.

- 패러티를 포함한 키 길이가 192 비트이면 (즉,3키 3중 DES), 암호화에 사용되는 3개의 독립적인 DES 키를 표현한다
-구현시 키의 패러티 비트는 무시하여야 함(즉, 체크하지 말아야 함)
- 그러나, DESede의 KeyGenerators 올바른 패러티를 갖는 키를 출력하여야 함
o 안전도관련 코멘트 :
- "Attacking Triple Encryption 주장
· 3키 3중 DES를 깨기 위해서는 약 2^{108}단계이면 충분하다
· 단독 DES 연산수에 집중하고, 다른 연산들이 매우 빠르다고 가정한다면, 2^{90} 단계이면 충분하다
- 이 보다 더 효과적 공격은 2키 3중 DES에 대항하도록 활용 가능(단지 Backward 호환성을 위해 사용될 경우)

■ DESX
o 설계자 : Ron Rivest
o 설명 :

$$EDESX[K, K1, K2](P) = EDES[K](P \oplus K1) \oplus K2$$
$$DDESX[K, K1, K2](C) = DDES[K](C \oplus K3) \oplus K2$$

- 만일 K, K1과 K2가 아래와 같이 서브키로 코드화된다면, 암호화와 복호화는 다음과 같이 정의된다.
- 만일 사용자 키 길이가 24 바이트이면, 첫 8 바이트는 DES 연산에 사용된 키 K를 표현하고, 8 바이트의 두 개의 연속적 블록은 "whitening" 키 K1 과 K2를 표현한다.
- 만일 사용자 키 길이가 24 바이트이면, 첫 8 바이트는 DES 연산에 사용된 키 K를 표현하고, 두 번째 8 바이트는 K와 K1으로 부터 유도된 "whitening" 키 K1 과 K2를 표현한다.
o 참고문헌 :
[정의] Mark Riordan, Subject: Re: Ladder DES. Posting to Usenet newsgroup sci.crypt, 1 Mar 1994. (Message-ID: <2ku9uc$sr8@msuinfo.cl.msu.edu>)

[분석] Joe Kilian, Phillip Rogaway, "How to protect DES against exhaustive key search," Earlier version in Advances in Cryptology - Crypto '96, Volume l109 of Lecture Notes in Computer Science (N. Koblitz, ed.), pp. 252-267. Springer-Verlag, 1996.
http://wwwcsif.cs.ucdavis.edu/~rogaway/papers/desx.ps
[정보] U.S. National Institute of Standards and Technology, NIST FIPS PUB 46-2 (supercedes FIPS PUB 46-1), "Data Encryption Standard", U.S. Department of Commerce, December 1993.
http://www.itl.nist.gov/div897/pubs/fip46-2.htm
[정보] Bruce Schneier, "Chapter 12 Data Encryption Standard," Applied Cryptography, Second Edition, John Wiley & Sons, 1996
[분석] J. Kelsey, B. Schneier, D. Wagner, " Related-Key Cryptanalysis of 3-WAY, Bihm-DES, CAST, DES-X, NewDES RC2, and TEA", ICICS '97 Proceedings, Springer-Verlag, November 1997.
http://www.counterpane.com/related-key_cryptanalysis.html
o 키 길이 : 128 또는 192 비트; 디폴트 192 비트(코드화 시)
※ 효과적 키 길이 : 안전도관련 코멘트 참조
o 블록 크기 : 8 바이트
o 코멘트:
- 구현시 키의 패러티 비트는 무시하여야 함
- DESX의 KeyGenerators는 완전 난수키를 생성하여야 함(DES의 취약키를 피할 수 있도록)
- 16바이트 키 경우, K2를 생성하는 해쉬과정의 입력은 패러티 조정 없이 원래 사용자 키이다.
- DESX를 위한 공식적인 참조 구현값은 없다

(Mark Riordan의 C 구현과 공식 DES 참조 데이터에 기반하여 手작업)
키 = <0123456789ABCDEF1011121314151617>
평문 = <445535864657378>
암호문 = <D8FA5084FAD4B35C>
키 = <0101010101010101010123456789ABCDEF1011121314151617>
평문 = <94DBE082549A14EF>
암호문 = <9011121314151617>
o 안전도 관련 코멘트 :
- “How to protect DES against exhaustive key search"에서 DES를 위한 Black Box 모델로 2^{118} 작업인자가 필요함을 증명
· 이 공격은 키 보수특성을 제의하고는 DES의 가능한 어떠한 취약성도 고려하지 않음
- 특히, DESX는 DC 및 LC 관점에서 DES보다 더 안전하지는 않음
- DESX는 연관키 공격에 취약하므로, 강한 PRNG로 키를 생성하거나 비트의 소스가 <해쉬함수의 출력과 같이> 충분히 상관성이 없어야 함

■ DFC
o 설계자 : Henri Gilbert, Marc Girault, Philippe Hoogvorst, Fabrice Noilhan, Thomas Pornin, Guillaume Poupard, Jacques Stem, Serge Vaudenay
o 공개 시기 : 1998. 5.
o 참고문헌 :
[정의, 분석] Henri Gilbert, Marc Girault, Philippe Hoogvorst, Fabrice Noilhan, Thomas Pornin, Guillaume Poupard, Jacques Stern, Serge Vaudenay, Decorrelated Fast Cipher: an AES Candidate,
http://csrc.nist.gov/encryption/aes/round1/AESAlgs/DFC/report.pdf
[정비] Olivier Baudron, Henri Gilbert, Louis Granboulan, Helena Handschuh, Robert Harley, Antoine Joux, Phong Nguyen, Fabrice Noilhan, David Pointcheval, Thomas Pornin, Guillaume Poupard, Jacques Stem, Serge Vaudenay, "DFC Update," Submitted to the 2nd AES Conference.

http://csrc.nist.gov/encryption/aes/round1/conf2/papers/baudrpn2.pdf
[정보] The Decorrelated Fast Cipher Home Page,
http://www. dmi.ens.fr/~vaudenay/dfc.html.
[정보] Serge Vaudenay, "The Decorrelation Technique,"
http://www. dmi.ens.fr/~vaudenay/decorrelation html.
[분석] Lars Knudsen, Vincent Rijmen, "On the decorrelated fast cipher (DFC) and its theory," Presented at Fast Software Encryption '99, Rome.
[특허] Serge Vaudenay, Proc?é de decorreiation des donn?s, French patent application num. 96 13411. Requested on 04/11/1996. (Extension to other countries in process.)
[테스트]」 NIST, DFC Test Values,
http://www-08.nist.gov/encrytion/aes/round1/testvals/dfc-vals.zip
o 키 길이 : 최소 0, 최대 256 비트, 8 비트의 배수; 디폴트 128 비트
o 블록 크기 : 16 바이트
o 코멘트
- "DFC Update" 논문에서 부가적 매개변수들 기술
· (블록 크기 m 비트, 라운드 수 r, 그리고 키 생성 조정 s)는 나중에 첨가될 예정
· 향후, 일반화 버전 명은 "DFC-m[(r[,s])]"
(각 m 값은 정상적으로 다른 구현의 기반이 되고, 반면 r과 s의 변화는 단독적인 구현으로 조정 가능 또한 가변 키길이는 매개변수로 표현될 필요 없음)
- DFC-128(디폴트: r=8, s=4)에 대하여, "DFC Update"에서 제안된 변화된 키 생성알고리즘은 아직 정확히 정의되지 않았음
o 특허 현황:
- DFC 자체는 특허가 없으나, decorrelation 기술은 특허가 있음

■ Diamond2(라운드)

o 설계자 : Michael Paul Johnson

o 공개 시기 : 1995

o 참고문헌:

[정의] Michael Paul Johnson, “The Dimnond2 Block Cipher,"
ftp://ftp.replay.com/pub/replay/mirror/ftp.cryptography.org/libraries/dlock2.zip

[정보] Michael Paul Johnson, "Beyond DES: Data Compression and the MPJ Encryption Algorithm," Master's Thesis at the University of Colorado at Colorado Springs, 1989.
ftp://ftp.replay.com/pub/replay/mirror/ftp.cryptography.org/libraries/dlock2.zip
(Note : 이것은 Diamond의 초기버전임(Diamond2가 아님)

[구현, 테스트] Michael Paul Johnson, Diamond2 reference implementation (in C++),
ftp://ftp.replay.com/pub/replay/mirror/ftp.cryptography.org/libraries/dlock2.zip

o 매개변수:

- 정수 라운드 [creation/read no 디폴트]- 수행될 라운드 수 (최소 10)

o 키 길이 : 최소 8, 최대 65536,8의 배수 비트; 디폴트 128 비트

o 블록 크기 : 16 바이트

o 코멘트:

- "The Diamond2 Block Cipher" 권고 라운드 수를 명기하지 않고 단지 최소 라운드 수만을 권고

· 이러한 이유로, 라운드 매개변수를 강제적으로 만들어야 함

- "Diamond2 Lite" 변종은 표준 명을 가지고 있지 않음

■ E2

o 설계자 : Kazumaro Aoki, Masayuki Kanda, Tsutomu Matsumoto, Shiho Moriai, Kazuo Ohta, Miyako Ookubo, Youichi Takashima, Hiroki Ueda

o 공개 시기 : 1998. 6.

o 참고문헌:

[정의, 분석] Specification of E2 - a 128-비트 Block Cipher,
http://info.isl.ntt.co.jp/e2/E2spec.pdf

[정보] The E2 Home Page,
http://info.isl.ntt.co.jp/e2

[정보] Supporting Docunent on E2,(corrected version, April 16 1999)
http://info.isl.ntt.co.jp/e2/E2support.pdf

[정보] Kazumaro Aoki, Hiroki Ueda, “Optimized Software Implementations of E2," Submitted to the 2nd AES Conference.
http://csrc.nist.gov/encryption/aes/round1/conf2/papers/aoki.pdf

[정보] Kazumaro Aoki, Hiroki Ueda, "Optimized Software Implementations of E2." Revised April 15, 1999
http://info.isl.ntt.co.jp/e2/RelDocs/implE2.pdf

[정의, 분석] M. Kanda, Y. Takashima, T. Matsumoto, K. Aoki, K. Ohta, "A Strategy for Constructing Fast Round Functions with Practical Security against Differential and Linear Cryptanalysis," Presented at the 5th annual workshop on Selected Areas in Cryptography(SAC '98) in August, 1998.

[분석] Makoto Sugita, Kazukuni Kobara, Hideki Imai, "Pseudorandomness and Maximum Average of Differential Probability of Block Ciphers with SPN-Structures 1ike E2," to the 2nd AES Conference.
http://csrc.nist.gov/encryption/aes/round1/conf2/papers/sugita.pdf

[분석] Yuji Hod, Toshinobu Kaneko, "A study of E2 by higher order differential attack," Technical report of IEICE. ISEC98-39, Science University of Tokyo (in Japanese -brief English summary here).

[분석] Mitsuru Matsui, Toshio Tokita, "On cryptanalysis of a byte-oriented cipher." The 1999 Symposium on Cryptography and Information Security, SCIS99-W2-1.5(in Japanese - brief English summary here).
[분석] Mitsuru Matsui, Toshio Tokita, "Cryptanalysis of a Reduced Version of the Block Cipher E2," Fast Software Encryption '99 (March 1999), pp. 70-79 (abstract here).
[분석] NTT Laboratories, "Security of E2 aginst Truncated Differential Cryptanalysis (in progress)," April 15 1999.
http://info.isl.ntt.co.jp/e2/RelDocs/E2tmnc.pdf
[특허] NTT (assignee), "Data Randomize Device and Symmetric Cipher Devices (translated)," Japanese Patent Application JP 173672/1997.
[특허] NTT (assignee),[[need patent titles]]
Japanese Patent Application JP 013572/1998.
Japanese Patent Application JP 013573/1998.
Japanese Patent Application JP 153066/1998.
Japanese Patent Application JP 147479/1998.
[테스트] NIST, E2 Test Values,
http://www-08.nist.gov/encrytion/aes/round1/testvals/dfc-vals.zip

o 키 길이 : 128, 192 또는 256 비트; 디폴트 128 비트
o 블록 크기 : 16 바이트
o 특허 현황 :
-NTT has several patents pending on E2 (참고문헌 참조).

■ FROG

o 설계자 : Dianelos Georgoudis, Damian Leroux, Billy Sim? Chaves (TecApro International S.A.)

o 참고문헌:

[정의, 분석] Dianelos Georgoudis, Damian Leroux, Billy Sim? Chaves, The “FROG” Encryption Algorithm,
http://www.tecapro.com/aesfrog2.htm (PDF version).

[분석] F. Koeune, G.F. Piret, J.J. Quisquater, Our first few comments about FROG, August 1998.
http//www.dice.ucl.ac.be/crypto/CAESAR/frog.html

[분석] David Wagner, Niels Ferguson, Bruce Schneier, Crptanalysis of FROG, Corrected version, March 16, 1999. Submitted to the 2nd AES Conference.
http://www-08.nist.gov/encrytion/aes/round1/testvals/frog-vals.zip

o 매개변수 :

- 정수 블록크기 [creation/read, 디폴트 16] - (8에서 1소) 바이트에서 블록 크기
- 정수 라운드 [creation/read, 디폴트 8] - 수행될 라운드 수(최소 8)

o 키 길이 : 최소 40, 최대 1000,8의 배수 비트; 디폴트 128 비트

o 블록 크기 : 블록 크기 매개변수(바이트)가 부여

o 빠진 정보:

- 16 바이트보다 큰 블록 크기에 대한 참조구현값

o 안전도관련 코멘트 :

- "Cryptanalysis of FROG"는 취약키에 관한 다음 공격을 기술

· 차분공격을 위해서는 2^{58} 선택 평문과 아주 적은 시간이 필요(키 공간 약 2^{330})

· 선형 공격은 2^{56} 알려진 평문을 사용

· 암호문 단독 선형 공격은 2^{64} 암호문($2^{-31.8}$의 키공간) 사용

· 복호화 함수에 대한 차분 공격은 2^{36} 선택 암호문 필요($2^{-29.3}$의 키공간)

■ GOST

o 별명 :“GOST-28147-89"

o 공개 시기 : 1989

o 참고문헌 :

[정의] GOST, Gosudarstvennyi Standard 28147-89, “Cryptographic Protection for Data Processing Systems," overnment Committee of the USSR for Standards, 1989(in Russian).

[정의, 정보] Bruce Schneier, "Section 14.l GOST," Applied Cryptography, Second Edition, John Wiley & Sons,1996.

[정보] J. Pieprzyk, L. Tombak, “Soviet Encryption Algorithm," Preprint 94-10, Department of Computer Science, The University of Wollongong, 1994.
ftp://ftp.cs.uow.edu.au/pub/papers/1994/tr-94-10.ps.Z

[분석] Markku-Juhani Saarinen, A chosen key attack against the secret S-boxes of GOST,
http://www.jyu.fi/~mjos/gost-cka.ps

[분석] C. Charnes, L. O'Connor, J. Pieprzyk, R. Safavi-Naini, Y. Zheng, "Further Comments on Soviet encryption algorithm,"Advances in Cryptology - EUROCRYPT '94 Proceedings, Volume 950 of Lecture Notes in Computer Science (A. De Santis, ed.), pp. 433-438. Springer Verlag,1995.

[분석] C. Charnes, L. O'Connor, J. Pieprzyk, R. Safavi-Naini, Y. Zheng, "Further comments on GOST encryption algorithm," Preprint 94-9, Department of Computer Science, The University of Wollongong, 1994.
ftp://ftp.cs.uow.edu.au/pub/papers/1994/tr-94-9.ps.Z

[분석] John Kelsey, Bruce Schneier, David Wagner, "Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and triple-DES," Advances in Cryptology - Crypto '96 Proceedings. Springer-Verlag, August 1996.

http://www.cs.berkeley.edu/~daw/papers/keysched-crypto96.ps
o 매개변수 :
- 바이트[ ][ ] s-box [write only, 디폴트 Applied Cryptography에 주어짐] - 이 암호알고리즘 예에서 사용하는 S-box
- s-box[i-1][j]는 입력값 j에 대한 s-box의 출력 i를 표현
- 구현에서 이 매개변수를 구성하는 배열의 내용을 복사할 수도 안할 수도 음
- 만일 그런 어떠한 배열이 연속적으로 변한다면, 암호알고리즘의 출력은 정의되지 않음
· 그러므로 Caller는 이 배열이 신뢰되지 않는 코드에 접근할 수 없다는 참조값을 만들 책임이 있음
- 가능하다면 이 매개변수를 구성할 때, 현재 키와 피드백 벡터를 재구성할 것임
o 키 길이 : 256 비트
o 블록 크기 : 8 바이트
o 빠진 정보 : 참조 구현값
o 안전도관련 코멘트 :
- “A chosen key attack against the secret S-boxes of GOST”는 약 2^{32} 암호화로 S-box를 복구하는 방법을 기술
· 주안점은 S-Box가 안전하다는 가정에서 tamperproof hardware에 관한 것임
· S/W 구현에서 어떠한 경우에서도 공개되었다고 가정될 수도 있음

■ HPC
o 설계자 : Rich Schroeppel
o 별명 : “Hasty pudding"
o 참고문헌 :

[정의] Rich Schroeppel, Hilarie Orman, Specification for the Hasty Pudding Cipher, http://www.cs.arizona.edu/~rcs/hpc/hpc-spec
[정보, 테스트] Rich Schroeppel, The Hasty Pudding Cipher page, http://www.cs.arizona.edu/~rcs/hpc/
[분석] David Wagner, Equivalent keys for HPC, Rump session talk at the 2nd AES Conference.
http://www.cs.berkeley.edu/~daw/papers/hpc-aes99-slides.ps
[분석] Carl D'Halluin, Gert Bijnens, Bart Preneel, Vincent Rijmen, Equivalent keys of HPC, Katholieke Universiteit Leuven, ESAT-COSIC.
http://www.esat.kuleuven.ac.be/~rijmen/pub99.html
[정보] Rich Schroeppel, “Tweaking the Hasty Pudding Cipher,"
http://www.cs.ahzona.edu/~rcs/hpc/tweak
[정보] Rich Schroeppel, “The Hasty Pudding Cipher: One Year Later," June 12, 1999.
http://www.cs.arizona.edu/~rcs/hpc/hpc-oneyearlater
[테스트] NIST, HPC Test Values,
http://www-08.nist.gov/encrytion/aes/round1/testvals/hpc-vals.zip
o 매개변수 :
- 정수 블록 크기 [cneation/read, 디폴트 16] - 바이트에서 블록 크기(최소1)
- 정수 백업[creation/read, 디폴트 0] - 속도 비용 측면에서(최소 0) 암호알고리즘을 보수적 만들려면 매개변수를 증가시킬 수 있다.
- 1ong[ ] spice [write, 디폴트 all-zeroes] - diversifier를 포함하는 8개의 64비트 워드 배열
- 구현에서 이 매개변수를 구성하는 배열의 내용을 복사할 수도 안 할 수도 있다.
· 만일 어떠한 배열이 부수적으로 변한 후, 매개변수가 즉시 다시 구성되지 않는다면 암호알고리즘의 출력은 정의되지 않는다
· 그러므로 Caller는 이 배열이 신뢰되지 않는 코드에 접근할 수 없다는 참조값을 만들 책임이 있음

- 매개변수를 구성할 때, 현재 키와 피드백 벡터를 재구성하지 말아야함
o 키 길이 : 최소 0, 최대 65536 비트; 디폴트 128 비트
o 블록 크기:
- 블록크기 매개변수에 의해 바이트로 주어짐
· Note : HPC가 8의 배수 비트를 지원하지 않으면, JCE API도 lwdnjs하지 않음.
o 안전도관련 코멘트 :
- "Equivalent keys of HPC"에서 D'Halluin et al은 128비트 키에 대한 공격을 기술
· 2^{89}기대 작용인자와 1/ 2^{56}의 키공간
· 192비트, 256비트 키로 확장하여도 비슷한 결과
· 다른 키길이(56비트 포함)도 취약할 것으로 보임
· "Tweaking the Hasty Pudding Cipher,"는 이 문제를 정정하려하였으나 정정본은 아직 SCAN 명을 받지 못함
- 디폴트 모두 'O' spice 값은 암호알고리즘이 많이 작용하여도 암호적 효과가 없음

■ ICE
o 설계자 : Matthew Kwan
o 참고문헌:
[정의, 테스트] Matthew Kwan, “The Design of the ICE Encryption Algorithm," Proceedings of Fast Software Encryption - Fourth International Workshop, Haifa, Israel, pp. 69-82. Springer-Verlag,1997.
http://www.darkside.com.au/ice/paper.html
[정보, 구현] The ICE Home Page,
http://www.darkside.com.au/ice/

[분석] Matthew Kwan, Cryptanalysis of ICE,
http://www.darkside.com.au/ice/cryptanalysis.html
[분석] B. Van Rompay, Lars Knudsen, Vincent Rijmen, "Differential cryptanalysis of the ICE encrytion algorithm," Fast Software Encryption, Volume 1372 of Lecture Notes in Computer Science (S. Vaudenay, ed.), pp. 270-283. Springer-Verlag, 1998.
ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/vrompay/fse98.ps.gz
o 키 길이 : 최소 64,64의 배수 비트; 디폴트 128 비트
o 블록 크기 : 8 바이트
o 코멘트 :
- 키 길이는 "level" 매개변수를 정의
(Note : "Thin ICE" 변종은 포함되지 않음).
o 안전도관련 매개변수 :
- "Differential cryptanalysis of the ICE encryption algorithm"은 차분 공격을 기술
· 15라운드로 축소된 변종에 대한 공격 포함
(2^{56} work와 기껏해야 2^{56} 선택 평문)
· 논문 결론 :
▷ Keyed permutation은 차분 공격을 막지 못함
▷ 키 의존성이 높을수록 비록 분석이 복잡하기는 하나, 설계의도를 만족시키지는 못함
▷ 본 논문의 공격에서 최상의 3 라운드 반복 특성은 2^{-13} 확률을 가짐
※ LOKI91의 최상 3 라운드 특성 확률 : 2^{-16}
(4개의 12에서 8비트로 가는 S-box를 사용하는 유사한 블록암호알고리즘)
▷ 본 공격은 라운드 수가 32 또는 그 이상인 경우 실제적이지 못하지만 이 경우, ICE는 DES보다 느림(즉, 키 크기가 128비트 이상일 경우)

■ lDEA

o 설계자 : Xuejia Lai, James Massey

o 별명 : “OpenPGP.Cipher.1”

“1.3.6.1.4.1.188.7.1.1.1” is an 별명 to “IDEA/ECB”

“1.3.6.1.4.1.188.7.1.1.2” is an 별명 to “IDEA/CBC”

“1.3.6.1.4.1.188.7.1.1.3” is an 별명 to “IDEA/CFB”

“1.3.6.1.4.1.188.7.1.1.4” is an 별명 to “IDEA/OFB”

(source for OIDs)

o 참고문헌 :

[정의, 분석] .X. Lai, "On the design and security of block ciphers", ETH Series in Infonnation Processing(J.L. Massey, ed.), Vol.1, Hartung-Gorre Verlag, Konstanz Technische Hochschule (Zurich), 1992.

[정보, 분석] X. Lai, J.L. Massey, S. Munrphy, "Markov Ciphers and Differential Cryptanalysis," Advances in Cryptology - EUROCRYPT '91, Volume 547 of Lecture Notes in Computer Science (D.W. Davies, ed.), pp. 17-38. Springer-Verlag, 1991.

[정보] The IDEA Algorithm page,
http://www.ascom.ch/infosec/idea.html.

[정보] Bruce Schneier, “Section 13.9 IDEA," Applied Cryptography, Second Edition, John Wiley & Sons, 1996.

[[note: there is an error in the 설명; diagram is correct]]

[정보] A. Menezes, P.C. van Oorschot, S.A. Vanstone, “Section 7.6 IDEA," Handbook of Applied Cryptography, CRC Press,1997.

[분석] Joan Daemen, Ren'e Govaerts, Joos Vandewalle, "Weak Keys of IDEA," Advances in Cryptology - CRYPTO '93 Proceedings, Volume 773 of Lecture Notes in Cornputer Science (D. Stinson, ed.), pp. 224-231. Springer-Verlag, 1994.

http://www.esat.kuleuven.ac.be/~rijmen/downloadable/daemen/idea.w.ps.gz
[분석] Joan Daemen, Ren'e Govaerts of IDEA," ESAT-COSIC Technical Report 93/1, 1993.
http://www.esat.kuleuven.ac.be/~cosicart/ps/JD-9306.ps.gz
[분석] J. Borst, L. Knudsen, V. Rijmen, "Two attacks on reduced IDEA," Advances in Cryptology - EUROCRYPT '97 Proceedings, Volume 1233 of Lecture Notes in Computer Science (W. Fumy, ed.), pp. 1-13. Springer-Verlag, 1997.
ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/rijmen/idea.ps.gz
[분석] L. Knudsen, V. Rijmen, "Truncated Differentials of IDEA," ESAT-COSIC Technical Report 97-1.
ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/knudsen/idea_trunc.ps.Z
[분석] J. Kelsey, B. Schneier, D. Wagner, “Key-Schedule Cryptanalysis of 3-WAY, IDEA, G-DES, RC4, SAFER, and Triple-DES", Advances in Cryptology - Crypto '96 Proceedings, pp.237-251. Springer-Verlag, August 1996.
http://www.counterpane.com/key_schedule.html
[분석] J. Kelsey, B. Schneier, D. Wagner, C. Hall, “Side Channel Cryptanalysis of Product Ciphers", ESORICS '98 Proceedings pp. 97-110, Springer-Verlag, September 1998.
http://www.counterpane.com/side_channel.html
[분석] E. G. Giessmann, G. Lassmann, "A Side Channel Cryptanalysis if the IDEA Cipher", Submitted to CRYPTO '99.
[정보, 분석] Ascom Systec, Ltd. "Side Channel Attack Hardening of the IDEA *IDEA*^{TM} Cipher," Ascom Systec White paper(corrected version, May 1999).
http://www.ascom.ch/infosec/downloads/sidechannel.pdf

[구현, 테스트] Ascom Systec, Ltd. IDEA C Source Code and Test Data(corrected version, May 1999).
http://www.ascom.ch/infosec/downloads.html
o 키 길이 : 128 비트
o 블록 크기 : 8 바이트
o 안전도관련 코멘트 :
- IDEA는 키특성 분석에 취약
· 그러므로 강한 PRNG로 키를 생성하여 사용하거나, 비트의 소스가 <해쉬함수의 출력과 같이> 충분히 상관성이 없어야 함
o 특허 현황 :
- Ascom Systec Ltd.사(일본 포함)는 미국 및 유럽 9개국에서 특허

■ LOKI91
o 설계자 : Laurence Brown, Matthew Kwan, Josef Pieprzyk, Jennifer Seberry
o 공개 시기 : 1991-92
o 참고문헌 :
[정의, 분석] Laurence Brown, Matthew Kwan, Josef Pieprzyk, Jennifer Seberry, "Improving Resistance to Differential Cryptanalysis and the Redesign of LOKI," Advances in Cryptology - ASIACRYPT '91 Proceedings, Springer-Verlag, 1993, pp. 36-50.
[정보] Bruce Schneier, “Section 13.6 LOKI," Applied Cryptography, Second Edition, John Wiley & Sons, 1996.
[분석] Eli Biham, "New Types of Cryptmnlytic Attacks Using Related Keys," CS 753, Computer Science Department, Technion -- Israel Institute of Technology, September 1992.
http://www.cstechnionac.il/users/wwwb/cgi-bin/tr-get.cgi/1992/CS/CSp753.ps
[분석] Lars Knudsen, "Cryptanalysis of LOK191," Volume 718 of Lecture Notes in Computer Science, pp. 196-208. Springer-Verlag, 1992.

ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/knudsen/loki91.ps.Z

[분석] Lars Knudsen, "Block ciphers - analysis, design and applications," PhD. Thesis. DAIMI PB 485, Aarhus University, 1994

[분석] Toshio Tokita, Tohru Sorimachi, Mitsuru Matsui, "Linear cryptanalysis of LOKI and S^2DES, "Volume 917 of Lecture Notes in Computer Science, pp. 293-306. Springer-Verlag, 1994.

[분석] Lars Knudsen, M.J.B. Robshaw, "Non-linear Approximations in Linear Cryptanalysis," Volume l070 of Lecture Notes in Computer Science, pp. 224-236. Springer-Verlag, 1996.

ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/knudsen/nonlinear.ps.Z

[분석] Kouichi Sakurai, Souichi Furuya, "Improving Linear Cryptanalysis of LOKI91 by Probabalistic Counting Method," Volume ??? of Lecture Notes in Computer Science. Springer-Verlag, 1997.

[분석] Lars Knudsen, “New potentially weak keys for DES and LOKI," Advances in Cryptology - EUROCRYPT '94 Proceedings, Volume 950 of Lecture Notes in Computer Science (A. De Santis, ed.), pp. 419-424. Springer Verlag, 1995.

ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/knudsen/potential.ps.Z

o 키 길이 : 64 비트

o 블록 크기 : 8 바이트

o 안전도관련 코멘트 :

- LOKI91은 연관키 공격에 취약(약 2^{60} 연산)

· 그러므로 강한 PRNG로 키를 생성하여 사용하거나, 비트의 소스가 <해쉬함수의 출력과 같이> 충분히 상관성이 없어야 함

- 축소 라운드(12라운드까지)에서 선형근사 공격이 효과적
· 총 라운드 수 : 16
- 고정된 64-비트 키 길이는 키 검색 공격시 너무 짧음

■ LOKI97

o 설계자 : Laurence Brown, Josef Pieprzyk, Jennifer Seberry
o 공개 시기 : 1997
o 참고문헌 :
[정의, 분석] Laurence Brown, Josef Pieprzyk, Introducing the new LOKI97 Block Cipher,
http://www.adfa.oz.au/~lpb/research/loki97/loki97spec.ps
[정보, 구현] Laurence Brown, The LOKI97 Block Cipher page,
http://www.adfa.oz.au/~lpb/research/1oki97/
[분석] Vincent Rijmen, Lars Knudsen, Weaknesses in LOKI97,
ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/rijmen/loki97.pdf
[테스트] NIST, LOKI97 Test Values,
http://www-08.nist.gov/encrytion/aes/round1/testvals/loki97-vals.zip
o 키 길이 : 128, 192 또는 256 비트; 디폴트 128 비트
o 블록 크기 : 16 바이트
o 안전도관련 코멘트:
- "Weaknesses in LOKI97
· 차분 공격(2^{56} 선택 평문)
· 선형 근사 공격(2^{56}기지 평문)

■ MARS

o 설계자 : Carolynn Burwick, Don Coppersmith, Edward D'Avignon, Rosario Gennaro, Shai Halevi, Charanjit Jutla, Stephen M. Matyas Jr., Luke O'Connor, Mohammad Peyravian, David Safford, Nevenko Zunicof

o 참고문헌 :

[정의, 분석] Carolynn Burwick, Don Coppersmith, Edward D´Avignon, Rosario Gennaro, Shai Halevi, Charanjit Jutla, Stephen M. Matyas Jr., Luke O'Connor, Mohammad Peyravian, David Safford, Nevenko Zunicof, MARS - A candidate cipher for AES,

http://www.research.ibm.com/security/mars.html

[분석] Markku-Juhani Saarinen, Equivalent keys in MARS (18 August 1998 -- corrected version).

http://www.math.jyu.fi/~mjos/mars_eqk.ps

[분석] Scott Contini, Yiqun Lisa Yin, "On Differential Properties of Data-Dependent Rotations and Their Use in MARS and RC6," Submitted to the 2nd AES Conference.

http://csrc.nist.gov/encryption/aes/round1/conf2/papers/contini.pdf

[특허] [[need patent title and date]]

U.S. Patent Application: IBM application CR998021.

[테스트] NIST, MARS Test Values,

http://www-08.nist.gov/encrytion/aes/round1/testvals/mars-vals.zip

o 키 길이 : 최소 32, 최대 1248,32의 배수 비트; 디폴트 128 비트

o 블록 크기 : 16 바이트

o 안전도관련 코멘트:

- 1248-비트 동치키는 232로 대부분의 키 길이에서 찾을 수 있음

· 이것은 자체적으로 중요한 취약점은 아님

· 그러나, Markku-Juhani Saahnen는 640비트 보다 더 많은 키(more keying material)를 사용할 경우, 충돌회피쌍 해쉬함수는 암호화 키를 생성하는데 사용할 수도 있다고 제안

o 특허 현황 :

- IBM이 특허권 소유

■ MAGENTA

o 설계자 : Michael Jacobson Jr., Klaus Huber

o 참고문헌:

[정의, 분석] M.J. Jacobson Jr., K. Huber, The MAGENTA Block Cipher Algorithm, http://csrc.nist.gov/encryption/aes/round1/AESAlgs/MAGENTA/megenta.pdf

[분석] Eli Biham, Alex Biryukov, Niels Ferguson, Lars Knudsen, Bruce Schneier, Adi Shamir, "Cryptanalysis of Magenta," Distributed at the first AES conference, August 20, 1998.

http://www.counterpane.com/magenta.html

[특허] [[need patent title]]

German Patent DE 44 25 158 Al.[[need date]]

[테스트] NIST, MAGENTA Test Values,

http://www-08.nist.gov/encrytion/aes/round1/testvals/magenta-vals.zip

o 키 길이 : 128, 256, 또는 256 비트; 디폴트 128 비트

o 블록 크기 : 16 바이트

o 안전도관련 코멘트:

- "Cryptanalysis of Magenta"

· 2^{64} 선택 평문과 2^{64} 연산

· Notes : 주어진 암호문에서 두 개의 반을 swap, 결과를 재 암호화 그리고 다시 swap 하여 복호화할 수 있다.

※ 이 방법이 비록 키를 얻지 못할 지라도, 어떤 응용에서는 치명적일 수 있음

■ RC2

o 설계자 : Ron Rivest

o 참고문헌 :

[정의] Ron Rivest, “A description of the RC2(r) Encryption Algorithm," RFC 2268, March 1998.
[분석] L.R. Knudsen, V. Rijmen, R.L. Rivest, M.J.B. Robshaw, "On the design and security of RC2," Fast Software Encryption, Volume 1372 of Lecture Notes in Computer Science (S. Vaudenay, ed.), pp. 206-221. Springer-Verlag, 1998.
[분석] J. Kelsey, B. Schneier, D. Wagner, "Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA," ICICS '97 Proceedings, Springer-Verlag, November 1997.
http://www.counterpane.com/related-key_cryptanalysis.html
o 키 길이 : 최소 0, 최대 1024,8의 배수 비트; 디폴트 128비트
o 블록 크기 : 8 바이트
o 안전도관련 코멘트 :
- 연관키 공격에 취약
· 강한 PRNG로 키를 생성하거나 비트의 소스가 <해쉬함수의 출력과 같이> 충분히 상관성이 없어야 함

■ RC5
o 설계자 : Ron Rivest
o 별명 : “RC5-32”
o 참고문헌:
[정보] Ron Rivest, “The RC5 Encryption Algorithm," Dr. Dobb's Joumal, vo1. 20 no. 1, pp. 146-148. January 1995.
[정의] Ron Rivest, "The RC5 Encryption Algorithm," (revised 20 March 1997)
http://theory.lcs.mit.edu/~rivest/rc5rev.ps
[정의] Ron Rivest, "The RC5, RC5-CBC, RC5-CBC-Pad, and RC5-CTS Algorithms," RFC 2040, October 1996.

[정보] A. Menezes, P.C. van Oorschot, S.A. Vanstone, "Section 7.7.2 RC5," Handbook of Applied Cryptography, CRC Press, 1997.
[분석] B.S. Kaliski, Y.L. Yin, "On Differential and Linear Cryptanalysis of the RC5 Encryption Algorithm", Advances in Cryptology - CRYPTO '95, pp. 171-184. Springer-Verlag, 1995.
[분석] Lars Knudsen, W. Meier, "Improved differential attack on RC5," Advances in Cryptology - Crypto '96 Proceedings, Volume 1109 of Lecture Notes in Computer Science (N. Koblitz, ed.), pp. 216-228. Springer-Verlag, 1996.
ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/Knudsen/rc5.ps.Z
[분석] H. Heys, "Linearly Weak Keys of RC5," IEE Electronics Letters, vol. 33, no. 10, pp. 836-838, 1997.
http://www.engr.mun.ca/~howard/PAPERS/rc5_letter.ps
[분석] A. Biryukov, E. Kushilevitz, "Improved Cryptanalysis of RC5, "EUROCRYPT '98.
http://www.cs.technion.ac.il/~eyalk/alex.ps.Z
[분석] A. A. Selcuk, "New results in linear cryptanalysis of RC5," Fast Software Encryption - Fifth International Workshop, Paris, France, LNCS. Springer-Verlag, 1998.
[분석] H. Handschuh, "A Timing Attack on RC5," Workshop on Selected Areas in Cryptography - SAC '98, Workshop record, 1998. To be published by Springer-Verlag.
http://www.enst.fr/~handschu/rc5.ps
[분석] H. Heys, "A Timing Attack on RC5," Workshop on Selected Areas in Cryptography - SAC '98, Queen's University, Kingston, Ontario, Aug. 1998.
http://www.engr.mun.ca/~howard/PAPERS/rc5_timing.ps
o 매개변수 :

- 정수 라운드 [creation/read, 디폴트 12] - (최소 12, 2의 배수)
o 키 길이 : 최소 0 ; 최대 65536,8의 배수 비트; 디폴트 128비트
o 블록 크기 : 8 바이트
o 안전도관련 코멘트 : 다수
o 특허 현황 : RSA Data Security, Inc.

■ RC5-64
o 설계자 : Ron Rivest
o 참고문헌 : RC5-32 참조
o 매개변수:
- 정수 라운드 [creation/read, 디폴트 16] - (최소 16,2의 배수)
o 키 길이 : 최소 0; 최대 65536,8의 배수 비트; 디폴트 128 비트
o 블록 크기 : 16 바이트
o 특허 현황 : RSA Data Security, inc.

■ RC6aymon
o 설계자 : Ron Rivest Matthew Robshaw Raymond Sidney, Yiqun Lisa Yin
o 별명 : "RC6-32"
o 참고문헌 :
[정의, 분석] Ron Rivest, M.J.B. Robshaw, R. Sidney, Y.L. Yin, The RC6 Block Cipher,
http://theory.lcs.mit.edt/~rivest/rc6.pdf
[분석] Ron Rivest, Further notes on RC6,
http://theory.lcs.mit.edu/~rivest/rc6-notes.txt
[분석] Scott Contini, Yiqun Lisa Yin, "On Differential Properties of Data-Dependent Rotations and Their Use in MARS and RC6." Submitted to the 2nd AES Conference.
http://csrc.nist.gov/encryption/aes/round1/conf2/papers/contini.pdf

[특허] RSA Data Security (assignee), "Block Encryption Algorithm with Data-Dependent Rotations," U.S. Patent #5,724,428. March 3,1998 (filed November l, 1995).
[특허] RSA Data Security (assignee), "Block Encryption Algorithm with Data-Dependent Rotations," U.S. Patent Application 08/854,210. Filed April 21, 1997.
[특허] RSA Data Security (assignee), "Encryption Algorithm with Data-Dependent Rotations," U.S. Patent Application 09/094,649. Filed June 15, 1998.
[테스트] NIST, RC6 Test Values,
http://www-08.nist.gov/encrytion/aes/round1/testvals/ro6-vals.zip
o 매개변수:
- 정수 라운드 [creation/read, 디폴트 20] - (최소 8, 4의 배수)
o 키 길이 : 최소 0, 최대 2040,8의 배수 비트; 디폴트 128비트
o 블록 크기 : 16 바이트
o 특허 현황 : RSA Data Security, Inc.

■ RC6-64
o 설계자 : Ron Rivest, Matthew Robshaw Raymond Sidney, Yiqun Lisa Yin
o 참고문헌 : RC6 참조
o 매개변수:
- 정수 라운드 [creation/read, 디폴트 20] - (최소 8, 4의 배수)
o 키 길이 : 최소 0, 최대 2040,8의 배수 비트; 디폴트 128비트
o 블록 크기 : 32 바이트
o 빠진 정보 : 참조 구현값
o 특허 현황 : RSA Data Security, Inc.

■ RC6a

o 설계자 : Ron Rivest, Matthew Robshaw Raymond Sidney, Yiqun Lisa Yin

o 별명 : "RC6a-32"

o 설명 : AES의 첫 라운드 평가에 따라 RC6-32의 키생성알고리즘 부분 수정

o 참고문헌 :

[정의] Scott Contini, Ron Rivest, M.J.B. Robshaw, Y.L. Yin, Some Comments on the First Round AES Evaluation of RC6,

http://csrc.nist.gov/encryption/aes/round1/comments/990414-mrobshaw.pdf

<RC6 참조>

o 매개변수 :

- 정수 라운드 [creation/read, 디폴트 20]-(최소 8, 4의 배수)

o 키 길이 : 최소 0, 최대 992,8의 배수 비트; 디폴트 128비트

o 블록 크기 : 16 바이트

o 빠진 정보 : 참조 구현값

o 특허 현황 : RSA Data Security, Inc.

■ RC6a-64

o 설계자 : Ron Rivest, Matthew Robshaw Raymond Sidney, Yiqun Lisa Yin

o 설명 : AES의 첫 라운드 평가에 따라 RC6-64의 키생성알고리즘 부분 수정

o 참고문헌 :

[정의] Scott Contini, Ron Rivest, M.J.B. Robshaw, Y.L. Yin, Some Comments on the First Round AES Evaluation of RC6,

http://csrc.nist.gov/encryption/aes/round1/comments/990414-mrobshaw.pdf

<RC6 참조>

o 매개변수 :

- 정수 라운드 [creation/read, 디폴트 20]- (최소 8, 4의 배수)
o 키 길이 : 최소 0, 최대 1984,8의 배수 비트; 디폴트 128비트
o 블록 크기 : 32 바이트
o 빠진 정보 : 참조 구현값
o 특허 현황 : RSA Data Security, Inc.

■ Rijndae1
o 설계자 : Joan Daemen, Vincent Rijmen
o 설명 : 128비트 블록 크기, 라운드 수(128비트 키 : 10라운드, 192비트 키 : 12 라운드, 256비트 키 : 14라운드)
o 참고문헌 :
[정의, 분석] Joan Daemen, Vincent Rijmen, AES Proposal: Rijndael,
http://www.esat.kuleuven.ac.be/~rijmen/rijndael/Rijndaeldoc.pdf
(Note: 본 논문에는 오류 있음)
[분석] Joan Daemen, Vincent Rijmen, Annex to AES Proposal: Rijndael,
http://www.esat.kuleuven.ac.be/~rijmen/rijndael/PropCorr.PDF
[정보, 테스트] Joan Daemen, Vincent Rijmen, The Rijndael Page,
http://www.esat.kuleuven.ac.be/~rijmen/rijndael/
[테스트] NIST, Rijndael Test Values,
http://www-08.nist.gov/encrytion/aes/round1/testvals/rijndael-vals.zip
o 키 길이 : 128,192 또는 256 비트; 디폴트 128비트
o 블록 크기 : 16 바이트

■ Rijndael-192
o 설계자 : Joan Daemen, Vincent Rijmen

o 설명 : 192비트 블록 크기, 라운드 수(128비트 키, 192비트 키 : 12 라운드, 256비트 키 : 14라운드)
o 참고문헌 : Rijndael 참조
o 키 길이 : 128, 192 또는 256 비트; 디폴트 128 비트
o 블록 크기 : 24 바이트

■ Rijndael-256
o 설계자 : Joan Daemen, Vincent Rijmen
o 설명 : 256 비트 블록 크기. 라운드 수는 항상 14.

■ SAFER-K
o 설계자 : James Massey
o 공개 시기 : 1993. 12
o 설명 : 1993년 설계된 SAFER의 원본
o 참고문헌 :
[정의] Massey, J. L., "SAFER K-64: A Byte-Oriented Block Ciphering Algorithm", Fast Software Encryption (R. Anderson, ed.), Proceedings of the Cambridge Security Workshop, Cambridge, U.K., December 9-11, 1993, pp. 1-17. Volume 809 of Lecture Notes in Computer Science, Springer, 1994.
[정보] Massey, J. L., “SAFER K-64: One Year Later”, Preliminary manuscript of a paper presented at the K. U. Leuven Workshop on Cryptographic Algorithms, December 14-16, 1994. To be published in the Proceedings of this workshop by Springer.
[정보] A. Menezes, P.C. van Oorschot, S.A. Vanstone, “Section 7.7.1 SAFER." Handbook of Applied Cryptography, CRC Press, 1997.

[정보, 테스트, 구현] Richard De Moliner, SAFER toolkit V1.2. Includes C implementation, additional notes, test data, test program.
ftp://ftp.isi.ee.ethz.ch/pub/simpl/safer.V1.2.tar.Z
[분석] Lars Knudsen, "A Key-Schedule Weakness in SAFER K-64", Advances in Cryptology - Crypto '95 Proceedings, Volume 963 of Lecture Notes in Computer Science, Springer-Verlag, 1995.
ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/knudsen/tillaeg.ps.Z
[분석] C. Harpes, "A Generalization of Linear Cryptanalysis Applied to SAFER," Internal report, Signal and Information Processing Lab., Swiss Federal Institute of Technology, Zurich, March 9, 1995.
http://www.isi.ee.ethz.ch/~harpes/GLCsafer.ps
[분석] Lars Knudsen, T.A. Berson, "Truncated differentials of SAFER," Fast Software Encryption, Volume l039 of Lecture Notes in Computer Science (D. Gollmann, ed.), pp. 15-26. Springer-Verlag, 1996.
ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/knudsen/trunc_dif_saf.ps.Z
[분석] J. Kelsey, B. Schneier, D. Wagner, "Key-Schedule Cryptanalysis of 3-WAY, IDEA, G-DES, RC4, SAFER, and Triple-DES", Advances in Cryptology - Crypto '96 Proceedings, pp. 237-251. Springer-Verlag, August 1996.
http://www.counterpane.com/key-schedule.html
o 매개변수:
- 정수 라운드 [creation/read, 디폴트 0(키길이에 의존)] - (최소 6)
- 64비트 키 : 6라운드, 128비트 키 : 10라운드
o 키 길이 : 64 또는 128 비트; 디폴트 128 비트

o 블록 크기 : 8 바이트
o 안전도관련 코멘트 :
- Lars Knudsen의 키특성분석 공격 결과, SAFER-K는 SAFER-SK로 대체

■ SAFER-SK
o 설계자 : James Massey
o 공개 시기 : 1995. 9.
o 설명 : SAFER 키생성 알고리즘을 강화시킨 알고리즘
o 별명 : "OpenPGP.Cipher.6"는 "SAFER-SK(13)"의 별명
o 참고문헌 :
[정의 (키생성 확장)] Massey, J. L., "SAFER K-64: A Byte-Oriented Block Ciphering Algorithm", Fast Software Encryption (R. Anderson, ed.), Proceedings of the Cambridge Security Workshop, Cambridge, U.K., December 9-11, 1993, pp. 1-17. Volume 809 of Lecture Notes in Computer Science, Springer, 1994.
[정의] Massey, J. L., "Announcement of a Strengthened Key Schedule for the Cipher SAFER", September 9, 1995, (see file 'SAFER_SK.TXT' included in the SAFER toolkit, below).
[정보, 테스트, 구현] Richard De Moliner, SAFER toolkit V1.2. Includes C implementation, additional notes, test data, test program.
ftp://ftp.isi.ee.ethz.ch/pub/simpl/safer.V1.2tar.Z
[분석] C. Harpes, "A Generalization of Linear Cryptanalysis Applied to SAFER," Internal report, Signal and Information Processing Lab., Swiss Federal Institute of Technology, Zurich, March 9, 1995.
http://www.isi.ee.ethz.ch/~harpes/GLCsafer.ps

[분석] Lars Knudsen, T.A. Berson, "Truncated differentials of SAFER," Fast Software Encryption, Volume l039 of Lecture Notes in Computer Science (D. Gollmann, ed.), pp. 15-26. Springr-Verlag, 1996.
ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/knudsen/trunc_dif_saf.ps.Z
[분석] J. Kelsey, B. Schneier, D. Wagner, "Key-Schedule Cryptanalysis of 3-WAY, IDEA, G-DES, RC4, SAFER, and Triple-DES", Advances in Cryptology - Crypto '96 Proceedings, pp. 237-251. Springer-Verlag, August 1996.
http://www.counterpane.com/key-schedule.html
o 매개변수 :
- 정수 라운드 [creation/read, 디폴트 0(키길이에 의존)] - (최소 8)
- 64비트 키 : 8라운드, 128비트 키 : 10라운드
o 키 길이 : 64 또는 128 비트; 디폴트 128 비트
o 블록 크기 : 8 바이트

■ SAFER+
o 설계자 : James Massey, Gurgen Khachatrian, Melsik Kuregian
o 참고문헌 :
[정의, 분석] Prof. James Massey, Prof. Gurgen Khachatrian, Dr. Melsik Kuregian, Nomination of SAFER+ as Candidate Algorithm for the Advanced Encryption Standard (AES),
http://www.cylink.com/intemet/objects.nsf/refernce/safer/$file/safer.pdf
[분석] Prof. James Massey, "On the Optimality of SAFER+ Diffusion," Submitted to the 2nd AES Conference.
http://csrc.nist.gov/encryption/aes/round1/conf2/papers/massey.pdf
[분석] John Kelsey, Bruce Schneier, David Wagner, "Key Schedule Weaknesses in SAFER+," Submitted to the 2nd AES Conference.

http://csrc.nist.gov/encryption/aes/round1/conf2/papers/kelsey.pdf
[테스트] NIST, SAFER+ Test Values,
http://www-08.nist.gov/encrytion/aes/round1/testvals/saferpls-vals.zip
o 키 길이 : 128, 192 또는 256 비트; 디폴트 128 비트
o 블록 크기 : 16 바이트

■ Serpent
o 설계자 : Ross Anderson, Eli Biham, Lars Knudsen
o 참고문헌 :
[정의, 분석] Ross Anderson, Eli Biham, Lars Knudsen, Serpent: A Proposal for the Advanced Encryption Standard,
http://www.cl.cam.ac.uk/ftp/users/rjal4/serpent.pdf
[정보] Ross Anderson, Serpent home page,
http://www.c1.cam.ac.uk/~rjal4/serpent.html
[정보] Eli Biham, Serpent page at Technion University,
http://www.cs.technion.ac.il/~biham/Reports/Serpent/
[분석] Orr Dunkelman, "An Analysis of Serpent-p and Serpent-p-ns," Submitted to the 2nd AES Conference.
http://csrc.nist.gov/encryption/aes/round1/conf2/papers/dunkelmen.pdf
[특허] Ross Anderson, Eli Biham, Lars Knudsen, "Fast Block Cipher," U.K. Patent Application 9722798.9. Filed October 30, 1997.
[구현] Frank Stajano, Markus Kuhn, Serpent reference implementations (in C, Python and Ada),
http://www.cl.cam.ac.uk/~rjal4/serpent.html
[테스트] NIST, Serpent Test Values,
http://www-08.nist.gov/encrytion/aes/round1/testvals/serpent-vals.zip
o 키 길이 : 128, 192 또는 256 비트; 디폴트 128 비트

o 블록 크기 : 16 바이트
o 특허 현황 :
- 공공분야에서 사용할 때에는 제한 없음

■ SHARK-A
o 설계자 : Vincent Rijmen, Joan Daemen, Bart Preneel Antoon Bosselaers, Erik De Win
o 설명 : SHARK 아핀변환 변종
o 참고문헌 :
[정의, 분석] Vincent Rijmen, Joan Daemen, Bart Preneel, Antoon Bosselaers, Elik De Win, “The Cipher SHARK,”
ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/rijmen/shark/pater.ps.
[구현] Vincent Rijmen, Joan Daemen, Bart Preneel, Antoon Bosselaers, Erik De Win, SHARK reference implementation(in C),
ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/rijmen/shark/
[분석] Thomas Jakobsen, Lars Knudsen, "The Interpolation Attack on Block Ciphers," Proceedings of Fast Software Encryption '97.
ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/knudsen/interpol.ps
o 키 길이 : 128 비트
o 블록 크기: 8 바이트
o 빠진 정보 : 참조구현값
o 안전도관련 코멘트 :
- SHARK 변종(제안 SHARK에서 약간 변형됨)을 보간공격으로 분석하여 5라운드로 줄임(SHARK는 6라운드를 가짐)

■ SHARK-E

o 설계자 : Vincent Rijmen, Joan Daemen, Bart Preneel Antoon Bosselaers, Erik De Win

o 설명 : SHARK의 EXor 변종

o 참고문헌 :

[정의, 분석] Vincent Rijmen, Joan Daemen, Bart Preneel, Antoon Bosselaers, Elik De Win, “The Cipher SHARK,”

ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/rijmen/shark/pater.ps.

[구현] Vincent Rijmen, Joan Daemen, Bart Preneel, Antoon Bosselaers, Erik De Win, SHARK reference implementation(in C),

ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/rijmen/shark/

[분석] Thomas Jakobsen, Lars Knudsen, "The Interpolation Attack on Block Ciphers," Proceedings of Fast Software Encryption '97.

ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/knudsen/interpol.ps

o 키 길이 : 128 비트

o 블록 크기: 8 바이트

o 안전도관련 코멘트 :

- SHARK 변종(제안 SHARK에서 약간 변형됨)을 보간공격으로 분석하여 5라운드로 줄임(SHARK는 6라운드를 가짐)

■ SKIPJACK

o 설계자 : U.S. National Security Agency

o 공개 시기 : 1998. 6.

o 참고문헌:

[정의] U.S. National Institute of Standards and Technology, SKIPJACK and KEA Specifications May 1998.
http://csrc.nist.gov/encryption/skipjack-kea.htm
[분석] Eli Biham, Alex Biryukov, Orr Dunkelman, Eran Richardson, Adi Shamir, "Observations on the SkipJack Encryption Algorithm,"
http://www.cs.technonac.il/~biham/Reports/SkipJack/
[분석] Eli Biham, Alex Biryukov, Orr Dunkelman, Eran Richardson, Adi Shamir, “Cryptanalysis of Skipjack Reduced to 31 Rounds using Impossible Differentials,"
http://www.cstechnionac.il/users/wwwb/cgi-bin/tr-get.cgi/1998/CS/CS0947.ps
o 키 길이 : 80 비트
o 블록 크기 : 8 바이트
o 안전도관련 코멘트 :
-SkipJack-3XOR
· SKIPJACK 320개 XOR중 3개를 제거함
· (두 번째 16비트 워드를 뺀 나머지는 모든 같은)약 500개의 평문으로부터 나온 암호문 이용
· 총 걸린 시간은 약 백만 SKIPJACK 암호화와 같음(PC상에서 수 초)

■ SPEED-64
o 설계자 : Yuliang Zheng
o 공개 시기 : 1997. 12.
o 참고문헌 :
[정의, 분석] Yuliang Zheng, "The SPEED Cipher," Proceedings of Financial Cryptography - First International Conference F'97, Volume 1318 of Lecture Notes in Computer Science, pp. 71-89. Springer-Verlag
http://www.pscit.monash.edu.au/~yuliang/pubs/speedc.tar.Z

[정보, 구현, 테스트] Yuliang Zheng, SPEED reference implementation(in C), http://www.pscit.monash.edu.au/~yuliang/pubs/speedc.tar.Z
[분석] C. Hall, J. Kelsey, B. Schneier, D. Wagner, "Cryptanalysis of SPEED", Fifth Annual Workshop on Selected Areas in Cryptography, Springer-Verlag, August 1998. http://www.counterpane.com/speed-sac.html
o 매개변수 :
- 정수 라운드 [creation/read, 디폴트 64] - 수행 라운드 수 (최소 64, 4의 배수)
o 키 길이 : 최소 48, 최대 256, 16의 배수 비트; 디폴트 128 비트
o 블록 크기 : 8 바이트
o 안전도관련 코멘트 :
- 라운드 수를 줄인 SPEED 변종(28라운드)는 2^{31}선택 평문으로 공격

■ SPEED-128
o 설계자 : Yuliang Zheng
o 공개 시기 : 1997. 2.
o 매개변수 :
- 정수 라운드 [creation/read, 디폴트 64] - 수행 라운드 수 (최소 64, 4의 배수)
o 키 길이 : 최소 48, 최대 256, 16의 배수 비트; 디폴트 128 비트
o 블록 크기 : 16 바이트

■ SPEED-128
o 설계자 : Yuliang Zheng

o 공개 시기 : 1997. 2.
o 참고문헌 : SPEED-64 참조
o 매개변수 :
- 정수 라운드 [creation/read, 디폴트 64] - 수행 라운드 수 (최소 64, 4의 배수)
o 키 길이 : 최소 48, 최대 256, 16의 배수 비트; 디폴트 128 비트
o 블록 크기 : 32 바이트

■ SQUARE
o 설계자 : Joan Daemen, Vincent Rijmen
o 공개 시기 : 1997
o 참고문헌:
[정의, 분석] Joan Daemen, Lars Knudsen, Vincent Rijmen, “The Block Cipher Square," Fast Software Encryption, Volume 1267 of Lecture Notes in Computer Science(E. Biham, ed.), pp. 149-165. Springer-Verlag, 1997.
http://www.esat.kuleuven.ac.be/~rijmen/downloadable/square/fse.pdf
[정보, 구현] Joan Daemen, Lars Knudsen, Vincent Rijmen, The Square Page,
http://www.esat.kuleuven.ac.be/%7Erijmen/square/
[테스트] Paulo Barreto, Validation data set for Square v1.0,
http://www.esat.kuleuven.ac.be/~rijmen/downloadable/square/vdata
o 매개변수 :
- 정수 라운드 [creation/read, 디폴트 8] - 수행 라운드 수 (최소 8, 2의 배수)
o 키 길이 : 128 비트
o 블록 크기 : 16 바이트

■ TEA
o 설계자 : David Wheeler, Roger Needham
o 공개 시기 : 1994
o 참고문헌:
[정의, 구현] David Wheeler, R. Needham, "TEA a Tiny Encryption Algorithm," Fast Software Encryption, Cambridge Security Workshop Proceedings, Volume 809 of Lecture Notes in Computer Science (R. Anderson, ed.), pp. 363-366. Springer-Verlag, 1994.
http://www.c1.cam.ac.uk/ftp/users/djw3/tea.ps
[분석1 J. Kelsey, B. Schneier, D. Wagner, "Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA", ICICS '97 Proceedings, Springer-Verlag, November 1997.
http://www.counterpane.com/related-key_cryptanalysis.html
o 키 길이 : 128 비트
o 블록 크기 : 8 바이트
o 빠진 정보 : 참조 구현값
o 안전도관련 코멘트:

- 연관키 공격에 취약
(그러므로 강한 PRNG로 키를 생성하여 사용하거나, 비트의 소스가 <해쉬함수의 출력과 같이> 충분히 상관성이 없어야 함)

■ XTEA

o 설계자 : David Wheeler, Roger Needham
o 공개 시기 : 1994

o 참고문헌:
[정의, 분석, 구현] David Wheeler, R. Needham, Tea extensions,
http://www.c1.cam.ac.uk/ftp/users/djw3/xtea.ps
o 키 길이 : 128 비트
o 블록 크기 : 8 바이트

■ Twofish
o 설계자 : Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hal1, Niels Ferguson
o 공개 시기 : 1998
o 참고문헌 :
[정의, 분석] Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hal1, Niels Ferguson, "Twofish: A128-bit Block Cipher," Submitted to the lst AES Conference, 15 June 1998.
http://www.counterpane.com/twofish-paper.html
[정보, 테스트, 구현] Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson, The Twofish: A New Block Cipher Page,
http://www.counterpane.com/twofish.html
[분석] Niels Ferguson, "Upper Bounds on Differential Characteristics in Twofish," Twofish Technical Report #1, August 17, 1998.
http://www.counterpane.com/twofish-differential.html
[분석1 Doug Whiting, David Wagner, "Empirical Verification of Twofish Key Uniqueness Properties," Twofish Technical Repot #2, Septelnber 22, 1998.
http://www.counterpane.com/twofish-keys.html
[분석] Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hal1, Niels Ferguson, "On the Twofish Key Schedule," Twofish Technical Report #3, Fifth Annual Workshop on Selected Areas in Cryptography, Springer Verlag, August 1998.
http://www.counterpane.com/twofish-keysched.html

[정보] Bruce Schneier, Doug Whiting, "Improved Twofish Implementations," Twofish Technical Report #3, December 2, 1998.
http://www.counterpane.com/twofish-speed.html
[분석] Fauzan Mirza, Sean Murphy, "An Observation on the Key Schedule of Twofish," Submitted to the 2nd AES Conference.
http://csrc.nist.gov/encryption/aes/round1/conf2/papers/mirza2.pdf
[분석] Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall. Niels Ferguson, on the Key Schedule of Twofish," Twofish Technical Report #4, March 16, 1999.
http://www.counterpane.com/twofish-ks2.html
[테스트] NIST, Twofish Test Values,
http://www-08.nist.gov/encrytion/aes/round1/testvals/twofish-vals.zip
o 키 길이 : 최소 8, 최대 256, 8의 배수 비트; 디폴트 128 비트
o 블록 크기 : 16 바이트

[첨부2] 주요 참고 문헌

[1] E. F. Brickell, J. H. Moore, and M. R. Purtill, "Structure in the s-boxes of the DES," Advances in Cryptology - CRYPTO'86, vol. 263 of Lecture Notes in Computer Science, pp. 3--8, Springer-Verlag,1987.
[2] D. Chaum and J.-H. Evertse, "Cryptanalysis of DES with a reduced number of rounds sequences of linear factors in block ciphers," Advances in Cryptology - CRYPTO '85 proceedings, vol. 218 of Lecture Notes in Computer Science, pp. 192-211, Springer-Verlag, 1986.
[3] D. W. Davies, "Some regular properties of the 'Data Encryption Sandard' algorithm," Advances in Cryptology: Proceedings of CRYPTO'82, pp. 89-96, Plenum Press, 1982.
[4] D. W. Davies and G.I.P. Parkin, "The average cyclew size of the key stream in ounput feedback enciphement," Cryptography, vol.149 of Lecture Notes in Computer Science, pp. 263-279, Springer-Verlag, 1983.
[5] NL Davio, Y. Desmedt, and J.-J. Quisquater, "Propagation characteristics of DES," Advances in Cryptology: Proceedings of Eurocrypt '84, pp. 62-73, Springer-Verlag, 1985.
[6] M. E. Hellman, R. C. Merkle, R. Schroeppel, L. Washington, W. Diffie, S. Pohlig, and P. Schweutzer, "Results of an initial attempt to cryptanalyze the NBS Data Encryption Standard," Tech. Rep. SEL 76-042, 1976.
[7] B. S. Kaliski Jr., R. L. Rivest, and A. T. Sherman, "Is the Data Encryption Standard a group?," Advances in Cryptology - EUROCRYPT'85, vol. 219 of Lecture Notes in Computer Science, pp. 81-95, Springer-Verlag, 1986.
[8] B. S. Kaliski Jr., R. L. Rivest, and A. T. Sherman, "Is DES a pure cipher?," Advances in Cryptology - CRYPTO'85 proceedings, vol. 218 of Lecture Notes in Computer Science, pp. 212-226, Springer-Verlag, 1986.

[9] Ii-Jun Kim and T. Matsumoto, "Achieving higher success probability in time-memory trade-off cryptanalysis without increasing memory size," IEICE Trans. Fundamentals, vol. E82-A, no. 1, pp. 123-129, l998.
[10] R. C. Merkle and M. E. Hellman, "On the security of multiple encryption," Comm. of ACM, vol. 24, pp. 465-467, 1981.
[11] J. H. Moore and G. J. Simmons, "Cycle structure of the DES with weak and semi-weak keys," Advances in Cryptology - CRYPTO'86, vol. 263 of Lecture Notes in Computer Science, pp. 9-32, Springer-Verlag, 1987.
[12] I. Schaumuller-Bichl, "Cryptanalysis of the Data Encryption Standard by the method of formal coding," Cryptography, vol. 149 of Lecture Notes in Computer Science, pp. 235-255, Springer-Verlag, 1983.
[13l A. Shamir, "On the security of DES," Advances in Cryptology - CRYPTO'85, vol. 218 of Lecture Notes in Computer Science, pp. 280 -281, Springer-Verlag, 1986.
[14] M. Wiener, "Efficient DES key Search," 1993
ftp://ripem.msu.edu/pub/crypt/docs/des-key-search.ps.
[15] M. Wiener, "Efficient DES key search - an update," CryptoBytes, vol. 3, no. 2, pp. 6-8, 1997.
[161 C. M. Adams, "On immunity against biham and shamir's "differential cryptanalysis," Information Processing Letters, vol. 41, no. 2, pp. 77-80, 1992.
[17] C. M. Adams and S. E. Tavares, "Designing s-boxes for ciphers resistant to differential cryptanalysis," Proceedings of the 3rd Symposium on State and Progress of Research in Cryptography, pp. 181 - 190, 1993.
[18] I. Ben-Aroya and E. Biham, "Differential cryptanalysis of Lucifer," Journal of Cryptology, vol. 9, no. 1, pp. 21-34, 1996.
[19] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials," Advances in Cryptology - EUROCRYPT'99, vol. 1592 of Lecture Notes in Computer Science, pp. 12-23, Springer-Verlag, 1999.

[20l E. Biham, A. Biryukov, and A. Shamir, "Miss in the rniddle attacks on idea, khufu and khafre." Preproceedings of FSE '99, pp. 121 -137, 1999.
[21] H. Gilbert and P. Chauvaud, "A chosen plaintext attack of the 16-round khufu cryptosystem," Advances in Cryptology - CRYPTO'94, vol. 839 of Lecture Notes in Computer Science, pp. 259-268, Springer-Verlag, 1994.
[22] E. Biham and. A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," J · Cryptology, vol. 4, no. 1, pp. 3-72, 1991.
[23] E. Biham and A. Shamir, "Differential cryptanalysis of Feal and N-hash," Advances in Cryptology - EUROCRYPT'91, vol. 547 of Lecture Notes in Computer Science, pp. 1-16, Springer-Verlag, 1991.
[24] E. Biham and A. Shamir, "Differential cryptanalysis of Snefru Khafre, REDOC-II, LOKI, and Lucifer," Advances in Cryptology - CRYPTO'91, vol. 576 of Lecture Notes in Computer Science, pp. 156-171, Springer-Verlag, 1992.
[25l E. Biham and A. Shamir, Differential Cryptanalysis of the Data Encryption Standard. Springer-verlag, 1993.
[26] L. Brown, M. Kwan, J. Pieprzyk, and J. Seberry, "Improving resistance to differential cryptanalysis and the redesign of LOKI," Advances in Cryptology - ASIACRYPT'91, vol. 739 of Lecture Notes in Computer Science, pp. 36 - 50, Springer-Verlag, 1991.
[27] D. Coppersmith, "The data encryption standard (DES) and its strength against attacks," Research Report RC 18613 (81421)12/22/92, IBM Research Division T. J. Watson Research Center, Yorktown Heights, NY 10598, 1992.
[28] J. Daemen, R. Govaerts, and J. Vandewalle, "Cryptanalysis of 2.5 rounds of IDEA," ESAT-COSIC Report 94-1, Dept. of Electrical Engineering, Katholieke Universiteit Leuven, March 1994.

[29] H. M. Heys and S. E. Tavares, "Substitution-permutation networks resistant to differential and linear cryptanalysis," Joumal of Crypto1ogy, vol. 9, no. 1, pp. 1-19, 1996.
[30] L. R. Knudsen, "Iterative characteristics of DES ans s$^ 2$-des," Advances in Cryptology - CRYPTO'92, Lecture Notes in Computer Science 740, pp. 497-511, Springer-Verlag, 1992.
[31] L. R. Knudsen, "Truncated and higher order differential," Fast Software Encryption - Second Intemational Workshop, Lecture Notes in Computer Science l008, pp. 196-211, Springer-Verlag, 1995.
[32] L. R. Knudsen and V. Rijmen, "Truncated differential of IDEA," ESAT-COSIC Report 97-1, Dept. of Electrical Engineering, Katholieke Universiteit Leuven, 1997.
[33] X. Lai, J. L. Massey, and S. Murphy, "Markov ciphers and differential cryptanaysis," Advances in Cryptology - EUROCRYPT'91, vol. 547 of Lecture Notes in Computer Science, pp. l7-38, Springer-Verlag, 1991.
[34] X. Lai, "Higher order derivatives and differential cryptanalysis," Communications and Cryptography, pp. 227-233, Kluwer Acadernic Publishers, 1994.
[35] W. Meier, "On the security of the IDEA block cipher." Advances in Cryptology - EUROCRYPT'93, vol. 735 of Lecture Notes in Computer Science, pp. 371-385, Springer-Verlag, 1994.
[36l S. Moriai, T. Schimoyama, and T. Kaneko, "Higher order differential attack of a cast cipher," Fast Software Encryption - 5th International Workshop, FSE'98, vol. 1372 of Lecture Notes in Computer Science, pp. 17-31, Springer-Verlag, 1998.
[37] E. Biham, "On Matsui's linear cryptanalysis," Advances in Cryptology - EUROCRYPT'94, vol.950 of Lecture Notes in Computer Science, pp. 341-355, Springer-Verlag, 1994.
[38] U. Bl\"ocher and M Dichtl, "Problems with the linear cryptanalysis of DES using more than ane active s-box per round," Fast Software Encryption - 2nd International Workshop, FSE'94, vol. 1008 of Lecture Notes in Computer Science, pp. 265-274, Springer-Verlag, 1995.

[39] D. Davies and S. Murphy, "Pairs and triplets of des s-boxes," Journal of Cryptology, vol. 8, no. 1, pp. 1-25, 1995.
[40] C. Harpes and J. L. Massey, "Partitioning cryptanalysis," Fast Software Encryption - 4th International Workshop, FSE'97, vol. 1267 of Lecture Notes in Computer Science, pp. 13-27, Springer-Verlag, 1997.
[41] P. Hawkes and L.O'Connor, "On applying linear cryptanalysis of IDEA," in Advances in Cryptology - AISACRYPT'96, vol. 1163 of Lecture Note in Computer Science, pp. 105-115, Springer-Verlag, 1996.
[42] B. S. Kaliski Jr. and M. J. B. Robshaw, "Linear cryptanalysis using multiple approximations," Advances in Cryptology - CRYPTO'94, vol. 839 of Lecture Notes in Computer Science, pp.26-39, Springer-Verlag, 1994.
[43] B. S. Kaliski Jr. and M. J. B. Robshaw, "Linear cryptanalysis using multiple approximations and FEAL," Fast Software Encryption, vol. 1008 of Lecture Notes in Computer Science, pp. 249-264, Springer-Verlag, 1995.
[44] B. S. Kaliski Jr. and Y. L. Yin, "On differential and linear cryptanalysis of the RC5 encryption algorithm," Advances in Cryptology - CRYPTO'95, vol. 963 of Lecture Notes in Computer Science, pp. 171-184, Springer-Verlag, 1995.
[45] L. R. Knudsen and M. J. B. Robshaw, "Non-linear approximations in linear cryptanalysis," Advances in Cryptology -EUROCRYPT'96, vol. 1070 of Lecture Notes in Computer Science, pp. 224-236, Springer-Verlag, 1996.
[46] M Matsui, "Linear cryptanalysis method for DES cipher," Advances in Cryptology - EUROCRYPT'93, vol. 765 of Lecture Notes in Computer Science, pp. 386-397, Springer-Verlag,1994.

[47] M. Matsui, "The first experimental cryptanalysis of the data encryption standard," Advances in Cryptology - CRYPTO'94, vol. 839 of Lecture Notes in Computer Science, pp. 1-11, Springer-Verlag, 1994.
[481 M Matsui and A.Yamsagishi, "A new method for known plaintext attack of FEAL cipher," Advances in Cryptology -EUROCRYPT'92, vol. 658 of Lecture Notes in Computer Science, pp. 81-91, Springer-Verlag, 1993.
[49] K. Ohta and K. Aoki, "Linear cryptanalysis of the fast data encipherment algorithm," Advances in Cryptology - CRYPTO'94, vol. 839 of Lecture Notes in Computer Science, pp. 12-16, Springer-Verlag, 1994.
[50] T. Shimoyama and T. Kaneko, "Quadratic relation of s-box and its application to the linear attack of full round des," Advances in Cryptology - CRYPTO'98, vol. 1462 of Lecture Notes in Computer Science, pp. 200-211, Springer-Verlag, 1998.
[51] E. Biham "New types of cryptanalytic attacks using related keys," Journal of Cryptology, vol. 7, no. 4, pp. 229-246, 1994.
[52] J. Daemen, R. Govaerts, and J. Vandewalle, "Weak keys for IDEA," Advances in Cryptology - CRYPTO'93, vol. 773 of Lecture Notes in Computer Science, pp. 224-230, Springer-Verlag, 1994.
[53] J. Kelsey, B. Schneier, and D. Wagner, "Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES," in Cryptology - CRYPTO'96, vol. 1109 of Lecture Notes in Computer Science, pp. 237-251, Springer-Verlag, 1996.
[54l J. Kelsey, B. Schneier, and D. Wagner, "Related-Key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, New DES, RC2, and TEA," Information and Communications Security - ICICS'97, vol. 1334 of Lecture Notes in Computer Science, pp. 233-246, Springer-Verlag, 1997.

[55] L. R. Knudsen, "Cryptanalysis of LOKI," Advances in Cryptology - ASIACRYPT'91, vol. 739 of Lecture Notes in Computer Science, pp. 22 - 35, Springer-Verlag, 1991.
[56] L. R. Knudsen, "Cryptanalysis of LOKI91." Advances in Cryptology - AUSCRYPT'92, vol. 718 of Lecture Notes in Computer Science, pp. 196-208, Springer-Verlag, 1993.
[57] B. S. Kaliski Jr. and Y. L. Yin, "A key-schedule weakness in safer k-64," Advances in Cryptology - CRYPTO'95, vol. 963 of Lecture Notes in Computer Science, pp. 274 - 286, Springer-Verlag, 1995.
[58] K. Aoki and K. Ohta, "Differential-linear cryptanalysis of feal-8," Proceedings of SISC'95, pp. 24-27, 1995.
[59] B. den Boer, "Cryptanalysis of F.E.A.L.," Advances in Cryptology - EUROCRYPT'88, Lecture Notes in Computer Science, pp. 293-299, Springer-Verlag, 1988.
[60] J. Borst, "Differential-linear cryptanalysis of IDEA," ESAT-COSIC Report 96-2, Dept. of Electrical Engineering, Katholieke Universiteit Leuven, 1996.
[61] F. Chabaud and S. Vaudenay, "Links between differential and linear cryptanalysis," Advances in Cryptology - EUROCRYPT'94, vol. 950 of Lecture Notes in Computer Science, pp. 363 - 374, Springer-Verlag, 1995.
[62] T. W. Cusic and M. C. Wood, "The redoc-ii cryptosystem," Advances in Cryptology - CRYPTO'90, vol. 523 of Lecture Notes in Computer Science, pp. 545-563, Springer-Verlag, 1991.
[63] T. Jakobsen and L. R. Knudsen, "The interpolation attack on block ciphers," Fast Software Encryption - Fourth International Workshop, vol. 1267 of Lecture Notes in Computer Science, pp. 28-40, Springer-Verlag, 1997.
[641 S. K. Langford and M. E. Hellman, "Differential-linear cryptanalysis," in Advances in Cryptology - CRYPTO'94, vol. 839 of Lecture Notes in Computer Science, pp. 17-25, Springer-Verlag, 1994.

[65] P. C. Oorschot and M J. Wiener, "A known-Plaintext attack on two-key triple encryption," Advances in cryptology - EUROCRYPT'90, vol. 473 of Lecture Notes in Computer Science, pp. 318-325, Springer-Verlag, 1991.
[66] S. Vaudenay, "On the need for multipermutations: Cryptanalysis of MD4 and SAFER," Fast Software Encryption, vol. 1008 of Lecture Notes in Computer Science, pp. 286-297, Springer-Verlag, 1995.
[67] W. Meier and O. Staffelbach, "Nonlinearity criteria for cryptographic functions," Advances in Cryptology - EUROCRYPT'89, vol. 434 of Lecture Notes in Computer Science, pp. 549-562, Springer-Verlag, 1990.
[68] C. J. Mitchell, "Enumerating boolean functions of cryptographic significance," J. Cryptology, vol. 2, no. 3, pp. 155-170, 1990.
[69] K. Nyberg, "Differentially uniform mappings for cryptography," Advances in Cryptology - EUROCRYPT'93, vol. 765 of Lecture Notes in Computer Science, pp. 55-64, Springer-Verlag, 1994.
[70] K. Nyberg, "S-boxes with controllable linearity and differential uniformity," Fast Software Encryption, vol. 1008 of Lecture Notes in Computer Science, pp. 111-130, Springer-Verlag, 1995.
[71] L. O'Conner and J. D. Golic, "A unified markov approach to differential and linear cryptanalysis," Advances in Cryptology -ASIACRYPT'94, vol. 917 of Lecture Notes in Computer Science, pp. 387-397, Springer-Verlag, 1995.
[72] J. Pieprzyk and G. Finkelstein, "Towards effective nonlinear cryptosystem design," IEE Proceedings (Part E), vol. 135, pp. 325-335, 1988.
[73] B. Preneel, W. V. Leekwick, L. V. Linden, R. Govaerts, and J. Vandewalle, "Propagation characteristics of boolean functions," in Advances in Cryptology - EUROCRYPT'90, vol. 473 of Lecture Notes in Computer Science, pp. 161-173, Springer-Verlag, 1991.

[74] J. Seberry, X. M. Zhang, and Y. Zheng, "GAC - the criterion for global avalanche characteristics of cryptographic functions," of Universal Computer Science, vol. 1, no. 5, pp. 320-337, 1995.
[75l Y. X. Yang and B. Guo, "Further enumeration boolean functions of cryptographic significance," J. Cryptology, vol. 8, no. 3, pp. 115-122, 1995.
[76] A. M. Adams and S. E. Tavares, "Generating bent sequences," Discrete Applied Mathematics, vol. 39, pp. 155-159, 1992.
[77] R. Forr\'e, "The strict avalanche criterion: Spectral properties of boolean functions and an extended definition," Advances in Cryptology - CRYPTO'88, Lecture Note in Computer Science 403, pp. 450-468, Springer-Verlag, 1989.
[78] A. F. Webster and S. E. Tavares, "On the design of s-boxes," Advances in Cryptology - CRYPTO'85, Lecture Notes in Computer Science 218, pp. 523-534, Springer-Verlag, 1986.
[79] K. Nyberg, "Perfect nonlinear s-boxes," Advances in Cryptology - EUROCRYPT'91, Lecture Notes in Computer Science 547, pp. 378-386, Springer-Verlag, 1991[] T. Beth and C. Ding, "On almost perfect nonlinear permutations," Advances in Cryptology - EUROCRYPT'93, Lecture Notes in Computer Science 765, pp. 65-76, Springer-Verlag, 1994.
[80] M. Matsui, "New structure of block ciphers with provable security against differential and linear cryptanalysis," Fast Software Encryption, vol. 1039 of Lecture Notes in Computer Science, pp. 205-218, Springer-Verlag, 1996.
[81] K. Nyberg and L. R. Knudsen, "Provable security against a differential attack," Joumal of Cryptology, vol. 8, no. 1, pp. 27-37, 1995.
[82] J. Daemen, Cipher and Hash Function Design Strategies Based on Linear and Differential Cryptanalysis. Doctoral dissertation, K. Leuven, 1995.

블록암호알고리즘 안전성 평가항목 분석

1999년 12월 인쇄
1999년 12월 발행
● 발행인: 이 철 수
● 발행처: 한국정보보호센터
서울시 서초구 서초동 1321-6
동아타워 5층
● 인쇄처: 대도문화사
전화:(02) 2273-5496

<비매품>