<2002 -Seminar in Graduate Courses>

Cryptography and Cryptanalysis

0. Study Areas on Information Security
I. Parallel Stream Cipher
II. Hackings and Countermeasures to Smartcard

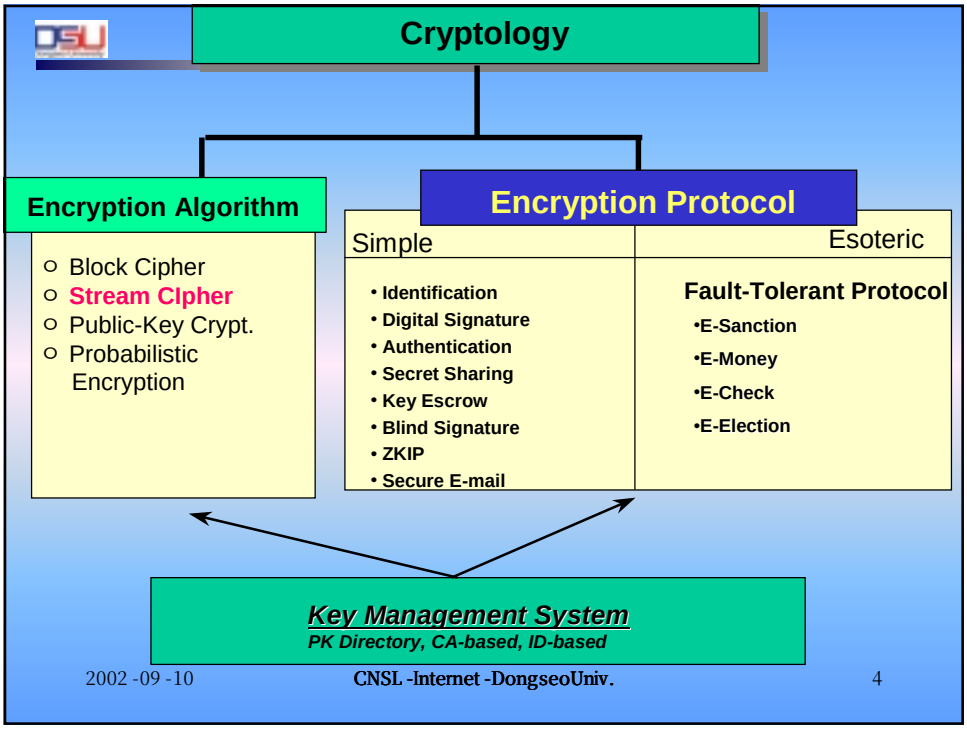
School of Internet Engineerings
Lee, Hoon -Jae
http://cg.dongseo.ac.kr/~hjlee
E -mail=hjlee@dongseo.ac.kr

0. Study Area on Information Security

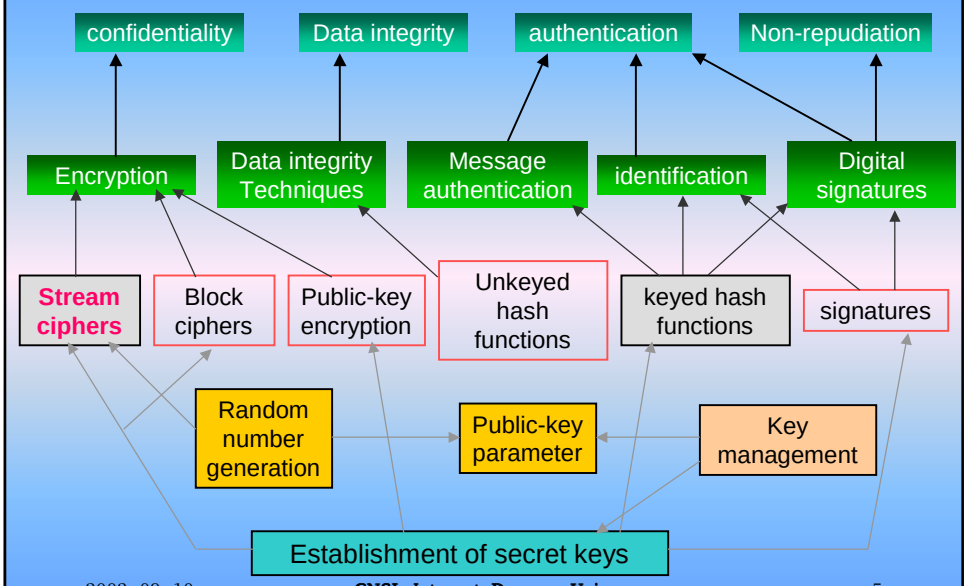
History in Cryptology

- BC Egyptian Pictograph(1822 decoded)
- BC60 Caeser cipher
- 1800. Vigenere cipher
- 1918. Vernam cipher = one-time pad
- 1917. Rotor Machine
- 1920. Enigma Machine, Hegelin Machine
- 1948. C.E.Shannon, "Secrecy System Theory"
- 1970. Europe, "Stream Cipher" design/applications
- 1970. Fiestel, "LUCIFER"
- 1975. IBM, "DES"
- 1976. Diffie & Hellman, "Public-Key Cryptosystems"
- 1978. Rivest, Shamir & Adleman, "RSA"
- 1980-90 A5, RC4, skipjack, AES(Rijndael), ECC

Cryptology



Cryptology



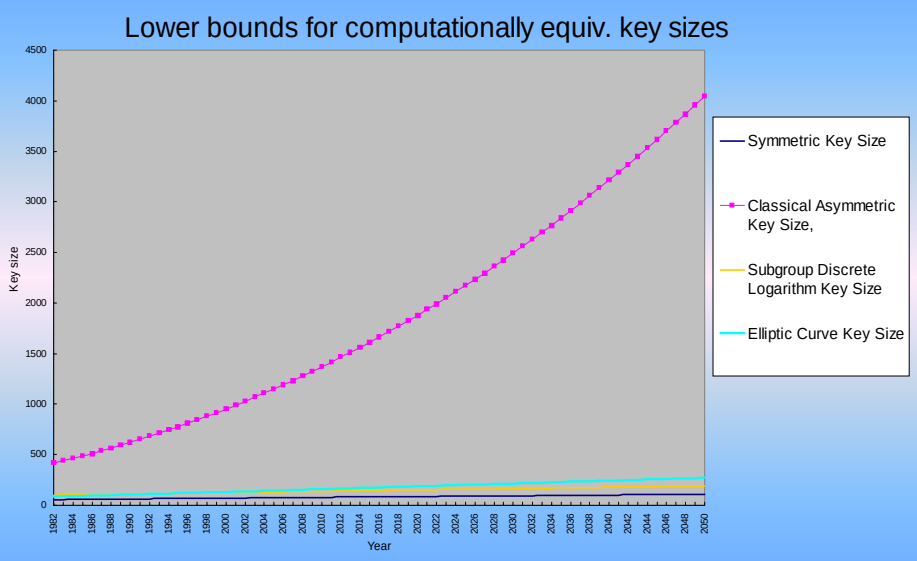
Secure Key Size

- Secure Key -Sizes vs. Years

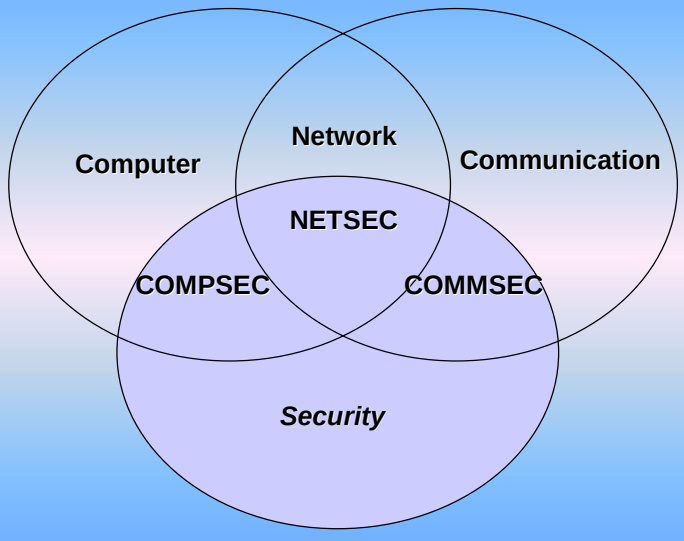
  Ref) A.K. Lenstra & E.R.Verheul, "Selecting Cryptographic Key Sizes" , PKC2000, Jan, 2000

Year	Symmetric	RSA/DH	Subgroup	Elliptic Curve	MIPS Year	H/W cost
2000	70	952	125	132	7.13×10^{9}	1.39×10^{8}
2005	74	1149	131	147	1.02×10^{11}	1.96×10^{8}
2010	78	1369	138	160	1.45×10^{12}	2.77×10^{8}
2020	86	1881	151	188	2.94×10^{14}	5.55×10^{8}

Secure Key Size



Information Security



I. Parallel Stream Cipher

Agenda

I. Introduction

II. Parallel -Structured LFSR

III. ***m*** -Parallel Nonlinear Combiners(4)

- ***m*-parallel** memoryless combiner
- ***m*-parallel** memory combiner
- ***m*-parallel** nonlinear filter
- ***m*-parallel** clock-controlled generator

IV. Design Example & Performance

V. Conclusion

I. Introduction

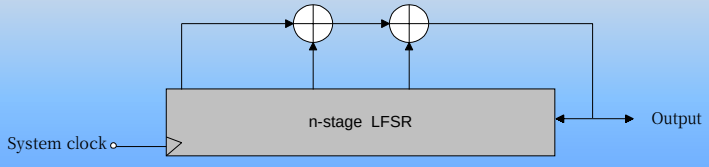
- International project
  - USA AES project → Rijnael
  - Japan CRYPTREC project
  - Europe NESSIE project : 10 Parts(New European Schemes for Signatures, Integrity, and Encryption) 2002. 12
- Requirements for communication
  - **High speed** communication
  - **High secure** encryption
  - **High reliable** communication for wired/wireless
  - **Bit-stream** Internet communication

⬇ ⬇ ⬇

- Proposal of four types of nonlinear combiner for parallel stream cipher & Design examples

II. Parallel-Strucured LFSR(1)

- Basic elements of stream cipher
  - LFSR, NFSR, Nonlinear Function
- LFSR(Linear Feedback Shift Register)
  - Primitive Poly. ⇨ Max. Period = 2^n - 1
  - External Clocks ⇨ 1-bit output/1-clock
  - Linear Output ⇨ Predictable !



II. Parallel-Strucured LFSR(2)

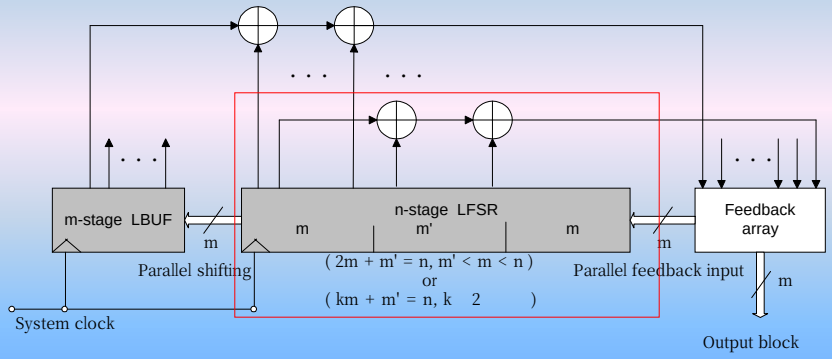
Parallel Stream Cipher

- Basic elements of parallel stream cipher
  - **PS-LFSR,** Serial/Parallel Converter
  - m-Parallel Nonlinear Functions
- Parallel-Shifting LFSR(PS-LFSR)
  - Security ⇨ the same as the LFSR
  - Peformance ⇨ m-bit output/1-clocks ⇨ m-times
    ($2 \le m \le n$)

[Ref] HJ Lee, SJ Moon" Parallel Stream Cipher for Secure High-Speed Communications" Signal Processing, Vol.82, No. 2, Feb. 2002.

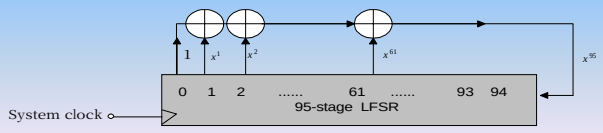
II. Parallel-Strucured LFSR(3)

PS-LFSR(Parallel-Shift LFSR)

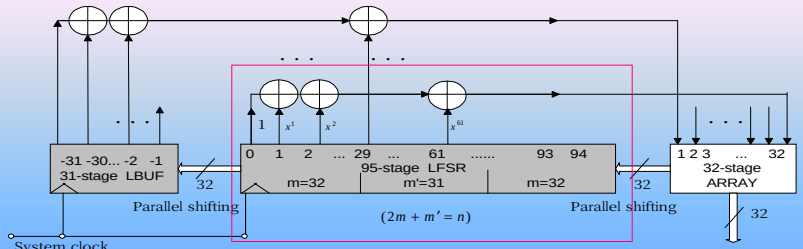


II. Parallel -Strucured LFSR(4)

Example: (n=95, m=32) PS -LFSR



a) Original LFSR : $p(x) = x^{95} + x^{61} + x^2 + x + 1$

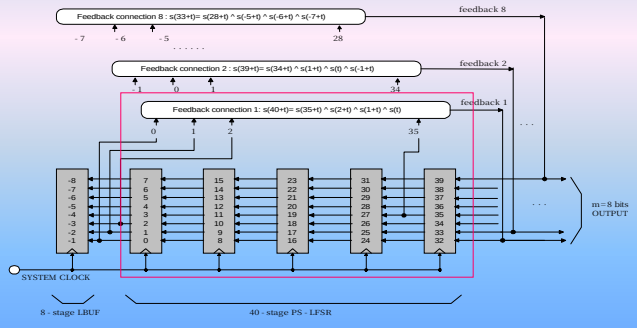


b) An Example PS - LFSR : $n = 95,\ m = 32,\ m' = 31$

II. Parallel -Strucured LFSR(5)

Example: (n=40, m=8) PS -LFSR

a) n=40 stage LFSR



b) (n,m)=(40,8) PS - LFSR

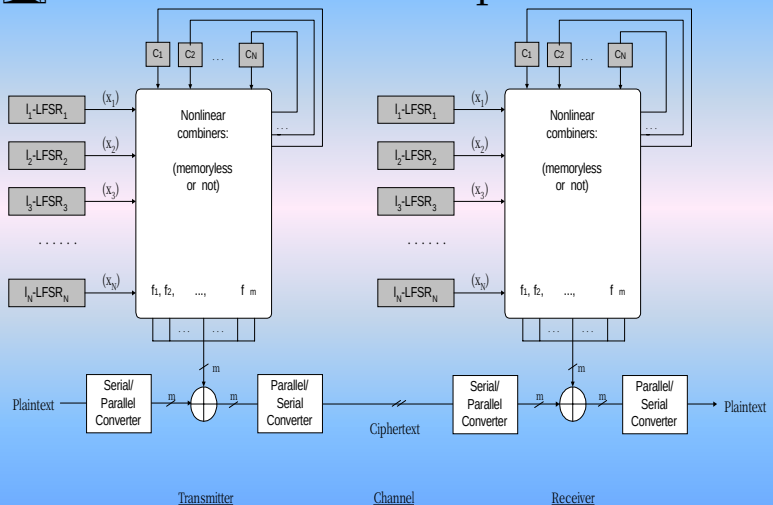
III. *m* -parallel nonlinear combiner(1)

Requirements of parallel stream cipher

- Stream Cipher ⇨ bit-serial comm., protection of error propagation
- Block Cipher ⇨ high-speed parallel processing
- Parallel Stream ⇨ hybrid

III. *m* -parallel nonlinear combiner(2)

m-Parallel Stream Cipher



III. *m* -parallel nonlinear combiner(3)

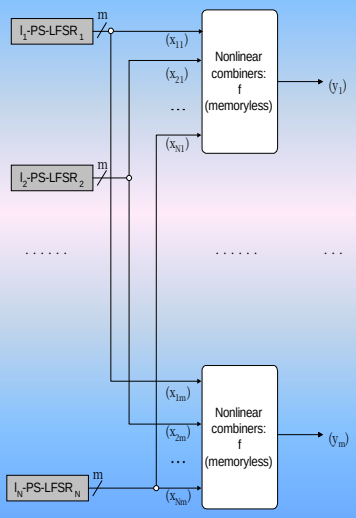
▩ *m*-parallel nonlinear combiners - requirements

(1) Randomness: Output sequence
(2) Period: Maximal
(3) Linear Complexity(LC)
(4) Correlation Immunity(CI): Input-Output
(5) Easy Implementation
(6) Easy Control of Secret Keys
 # (7) Correlation Immunity(CI): Output-Output

III. *m* -parallel nonlinear combiner(4)

1. *m* -parallel memoryless combiners

- Nxm input seq.: $(x_{11}),\ldots,(x_{Nm})$
- N PS-LFSR : $I_{1\sim N}$-$LFSR_{1\sim N}$
- m output seq.: $y_1,\ldots,y_m$
- m functions: $f_1,\ldots,f_m$



III. *m* -parallel nonlinear combiner(5)

m -parallel memoryless combiners

$$f_1(x_{11},x_{21},\ldots,x_{N1}) = a_{1,0} + (\sum_{i=1}^{N} a_{1,i}x_{i1}) + (\sum_{i,j} a_{1,ij}x_{i1}x_{j1})$$

$$+\ldots+a_{1,12..N}x_{11}x_{21}..x_{N1}$$

$$f_2(x_{12},x_{22},\ldots,x_{N2}) = a_{2,0} + (\sum_{i=1}^{N} a_{2,i}x_{i2}) + (\sum_{i,j} a_{2,ij}x_{i2}x_{j2})$$

$$+\ldots+a_{2,12..N}x_{12}x_{22}..x_{N2}$$

$$\ldots\ldots$$

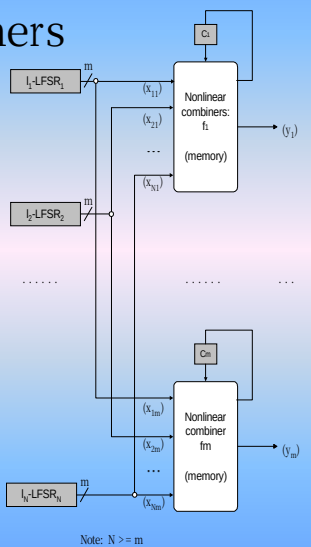
$$f_m(x_{1m},x_{2m},\ldots,x_{Nm}) = a_{m,0} + (\sum_{i=1}^{N} a_{m,i}x_{im}) + (\sum_{i,j} a_{m,ij}x_{im}x_{jm})$$

$$+\ldots+a_{m,12..N}x_{1m}x_{2m}..x_{Nm}$$

III. *m* -parallel nonlinear combiner(6)

2. *m* - parallel memory combiners

- Nxm input seq.: $(x_{11}),\ldots,(x_{Nm})$
- N PS-LFSR : $l_{1\sim N}$-$LFSR_{1\sim N}$
- m output seq.: $y_1,\ldots,y_m$
- m nonlinear combiners: $f_1,\ldots,f_m$
- m memory seq.: $c_1,\ldots,c_m$



III. *m* -parallel nonlinear combiner(7)

- *m* - parallel memory combiners

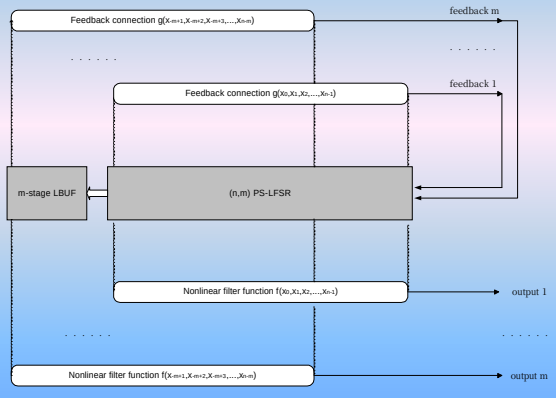
$$f_1(x_{11},x_{21},\ldots,x_{N1},c_{11},c_{12},\ldots,c_{1M_1}) = a_{1,0} + (\sum_{i=1}^{N} a_{1,i}x_{i1} + \sum_{i=N+1}^{N+m} a'_{1,i}c_{1i}) + (\sum_{i,j} a_{1,ij}x_{i1}x_{j1} + \sum_{i,j} a'_{1,ij}c_{1i}c_{1j} + \sum_{i,j} a''_{1,ij}x_{i1}c_{1j}) + \ldots + a_{1,12..N+M_1}x_{11}x_{21}..x_{N1}c_{11}c_{12}..c_{1M_1}$$

$$\ldots\ldots$$

$$f_m(x_{1m},x_{2m},\ldots,x_{Nm},c_{m1},c_{m2},\ldots,c_{mM_m}) = a_{m,0} + (\sum_{i=1}^{N} a_{m,i}x_{im} + \sum_{i=N+1}^{N+m} a'_{m,i}c_{mi}) + (\sum_{i,j} a_{m,ij}x_{im}x_{jm} + \sum_{i,j} a'_{m,ij}c_{mi}c_{mj} + \sum_{i,j} a''_{m,ij}x_{im}c_{mj}) + \ldots + a_{m,12..N+M_m}x_{1m}x_{2m}..x_{Nm}c_{m1}c_{m2}..c_{mM_m}$$

III. *m* -parallel nonlinear combiner(8)

- 3. *m* -parallel nonlinear filter functions

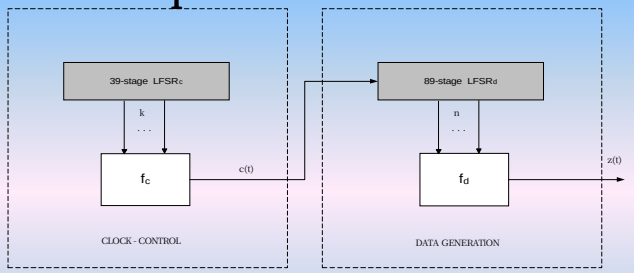


III. *m*-parallel nonlinear combiner(9)

- *m*-parallel nonlinear filter functions
  - *m*-parallel feedback connections :
    - $g(x_{-1},x_0,x_1,\ldots,x_{n-2})$
    - . . . . . .
    - $g(x_{-m+1},x_{-m+2},x_{-m+3},\ldots,x_{n-m})$
  - *m*-parallel nonlinear filter functions :
    - $f(x_{-1},x_0,x_1,\ldots,x_{n-2})$
    - . . . . . .
    - $f(x_{-m+1},x_{-m+2},x_{-m+3},\ldots,x_{n-m})$

III. *m*-parallel nonlinear combiner(10)

4. *m*-parallel clock-controlled gen.
   - Example :LILI-128



- One of NESSIE proposal, QUT, Austalia
- 39-stage $LFSR_c$
- 89-stage $LFSR_d$
- Clock-control function $f_c(t)$
- Nonlinear filter function $f_d(t)$

III. *m* -parallel nonlinear combiner(11)

LILI-128 stream cipher (2)

- Primitive polynomial of 39-stage $LFSR_c$

$$g_c(x) = x^{39} + x^{35} + x^{33} + x^{31} + x^{17} + x^{15} + x^{14} + x^2 + 1$$

- Primitive polynomial of 89-stage $LFSR_d$

$$g_d(x) = x^{89} + x^{83} + x^{80} + x^{55} + x^{53} + x^{43} + x^{39} + x + 1$$

- Clock-control function $f_c(t)$

$f_c(t) = 2\,c(12+t) + c(20+t) + 1$ on real field

- Nonlinear filter function $f_d(t)$

$f_d(0,1,3,7,12,20,30,44,65,80)$=6th order function

III. *m* -parallel nonlinear combiner(12)

LILI-128 stream cipher (3)

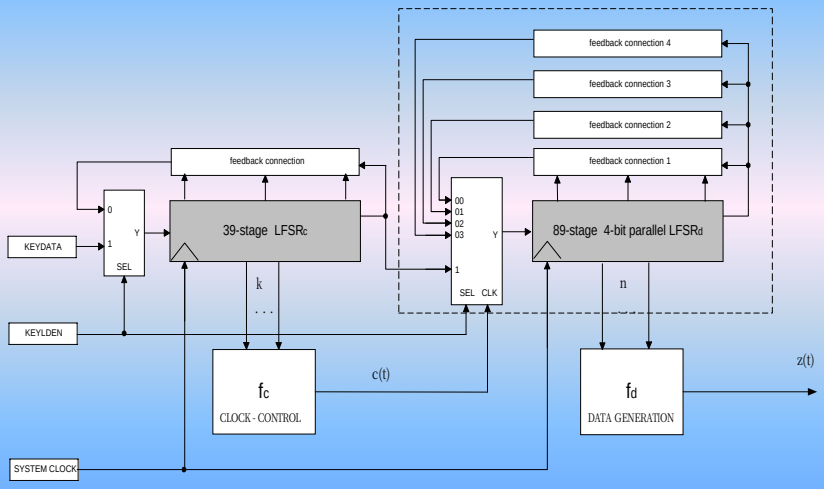
- Period: $P = (2^{39} - 1)(2^{89} - 1) \approx 2^{128}$
- Linear Complexity:

$$LC \geq \binom{L_d}{r} \cdot P_c = \binom{89}{6} \cdot (2^{39} - 1) \approx 2^{68}$$

- Nonlinear filter function $f_d(t)$: Table
  - Balanced, CI(3), Nonlinear Order=6
  - 10 inputs w/ 0,1,3,7,12,20,30,44,65,80

III. *m* -parallel nonlinear combiner(13)

Parallelization of clock-control $LFSR_d$



III. *m* -parallel nonlinear combiner(14)

$LFSR_d$'s feedback connections

- Feedback connection 1

$d[89+t]=d[88+t]\oplus d[50+t]\oplus d[47+t]\oplus d[36+t]\oplus d[34+t]\oplus d[9+t]\oplus d[6+t]\oplus d[t]$

- Feedback connection 2

$d[88+t]=d[87+t]\oplus d[49+t]\oplus d[46+t]\oplus d[35+t]\oplus d[33+t]\oplus d[8+t]\oplus d[5+t]\oplus d[-1+t]$

- Feedback connection 3

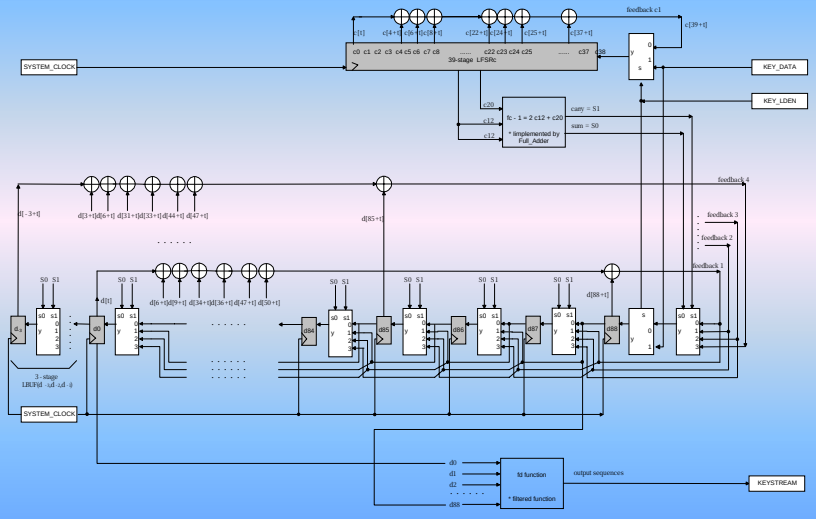
$d[87+t]=d[86+t]\oplus d[48+t]\oplus d[45+t]\oplus d[34+t]\oplus d[32+t]\oplus d[7+t]\oplus d[4+t]\oplus d[-2+t]$

- Feedback connection 4

$d[86+t]=d[85+t]\oplus d[47+t]\oplus d[44+t]\oplus d[33+t]\oplus d[31+t]\oplus d[6+t]\oplus d[3+t]\oplus d[-3+t]$

III. *m* -parallel nonlinear combiner(15)

Design of 1/2/3/4 -parallel $LFSR_d$

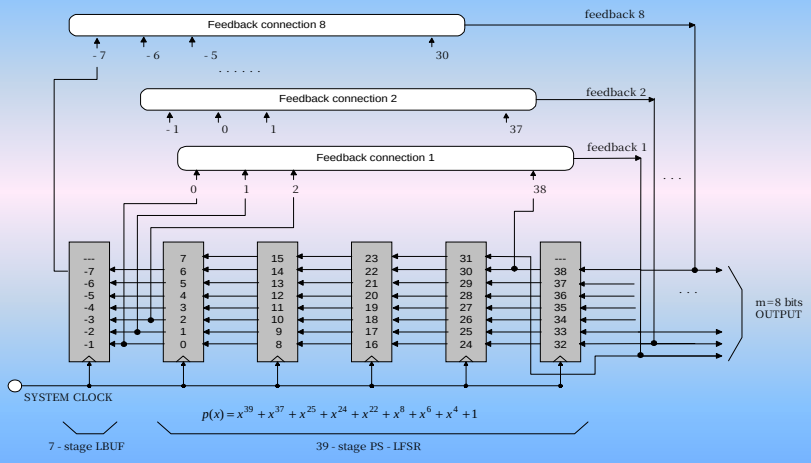


IV. Design Example & Performance(1)

- ALTERA's Max+plus II tools
- FPGA device: EPF10K20RC240-3
- System clock: 50MHz
  - Max. path delay < 20 ns
  - m=8 times high-speed
  - Hardware Complexity < 8 ?

Ⅳ. Design Example & Performance(2)

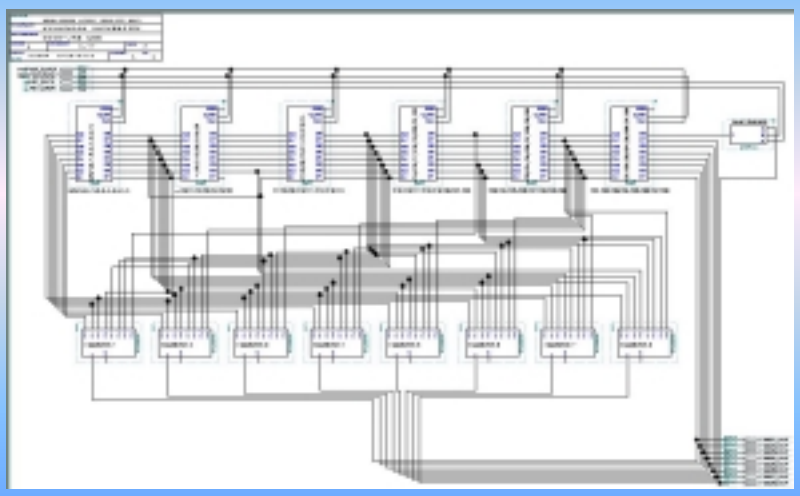
(n=39, m=8) PS -LFSR



b) (n,m)=(39,8) PS - LFSR

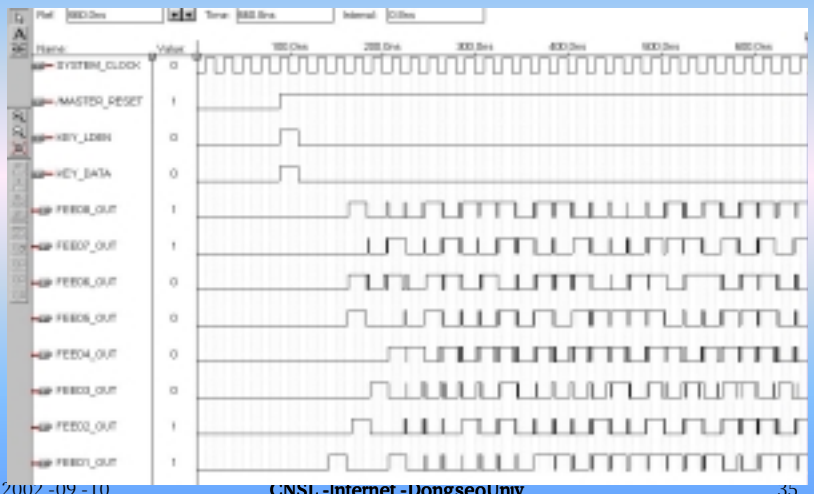
Ⅳ. Design Example & Performance(3)

(n=39,m=8)PS -LFSR: Graphic Design



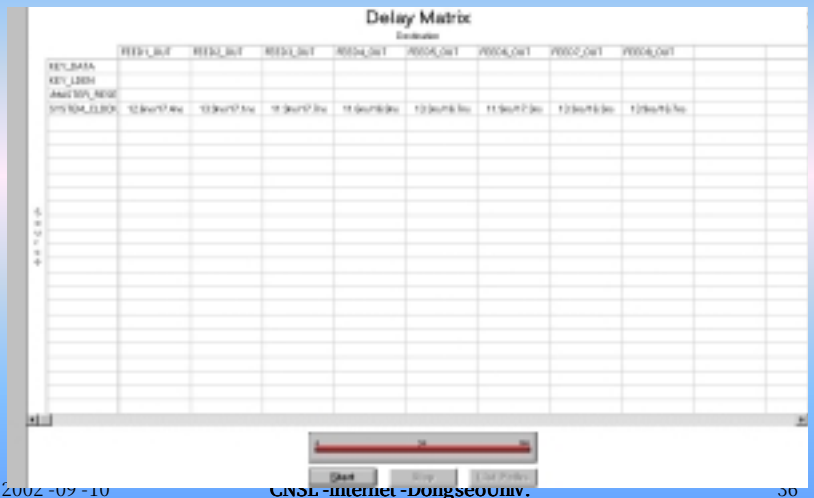
Ⅳ. Design Example & Performance(4)

- (n=39, m=8) PS -LFSR : Simulation



Ⅳ. Design Example & Performance(5)

- (n=39, m=8) PS -LFSR : Time Delay



Ⅳ. Design Example & Performance(6)

Performances of LFSR vs. PS-LFSR

Items	39-LFSR	(39,8) PS-LFSR
Period	$2^{39}-1$	$2^{39}-1$
Processing rate @ 50MHz system clock	50 Mbps	50 x 8 = 400 Mbps (Max. Delay = 17.3 ns)
Hardware Complexity [gates]	219 gates -39 D F/Fs -1 (2-1) MUX -7 XORs	401 gates (1.83 : 1) -46 D F/Fs -1 (2-1) MUX -56 XORs

Ⅴ. Conclusions

- *m*-parallel nonlinear combiners : 4 types
  1. ***m*-parallel** memoryless combiners
  2. ***m*-parallel** memory combiners
  3. ***m*-parallel nonlinear** filter functions
  4. ***m*-parallel** clock-controlled generators
     - **Examples: LILI-128**

⬇ ⬇ ⬇

- **Implementing FPGA: EPF10K20RC240-3**
  - ⇨ Hardware complexity: 2 times **(PS-LFSR)**
  - ⇨ Output Performance: m times **available**

II. Hackings and Countermeasures to Smartcard

발표 순서

- 서론
- 부채널공격 관련 안전성 평가기준
- 스마트카드에 대한 부채널 공격 분석
- DES에 대한 DPA 실험 및 분석
- DES에 대한 DPA 방어기술 제안
- 결론

서 론

❑ 스마트카드
- ➢ **공무원카드, 전자화폐, 교통카드, 의료카드, 신분카드** 등에 적용
- ➢ **한국전자통신연구원 제시 IT분야 30/40대 전략품목**
- ➢ 2002년까지, (국내) 수요 **1,000만개의** 카드 소지 유도 : 정보통신부
  (국제) 수요 **50억개 (70억불)**의 생산이 예측

❑ **최신 스마트카드 해킹 기술 분석**
- ➢ 미국 크립토리서치 사 Kocher에 의하여 제안
- ➢ **부채널공격 유형 분석** : 시간공격/결함주입공격/전력분석공격
- ➢ **전력분석공격 유형 분석** : SPA/DPA/IPA/HO -DPA 공격법
- ➢ **RSA 공격 유형 분석** : SEMD/MESD/ZEMD 공격법

❑ **DPA 실험 및 방어기법 제안**
- ➢ **DES 암호에 대한 DPA 실험 분석**
- ➢ **DES 암호에 대한 소프트웨어적인 방어기법 제안**

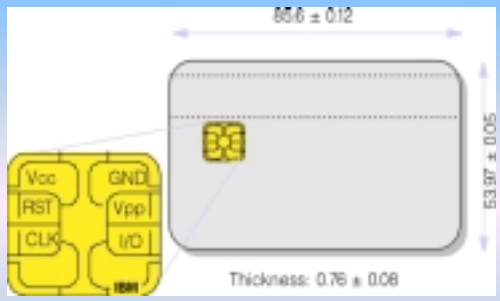
❑ 국내 스마트카드 평가기준서(PP, ST), 평가방법론(CEM)작성 기초자료
- ➢ 부채널 분석(Timming attack, Fault Insertion, Power attack 등)

부채널공격 관련 안전성 평가 기준

❑ 스마트카드 관련 안전성 평가 기준
- ➢ 국제공통기준(CC)
  - ✓ SCSUG -PP(ver 3.0)/EUROSMART -CPP
  - ✓ Fault Attack, Power Attack**에 대한 방어 요구**
- ➢ 미국 암호모듈 평가기준(CMVP, FIPS -140)
  - ✓ Fault Attack, Power Attack, Timing Attack, TEMPEST**에 대한 방어 요구**
- ➢ 유럽 차세대 암호 공모 과제(NESSIE)
  - ✓ Timing Attack, Power Attack **등 평가항목 포함**
- ➢ 일본 원천암호기술 공모 과제(CRYPTREC)
  - ✓ 암호 알고리듬 7개 분야의 후보에 대한 평가 진행 중

스마트카드에 대한 부채널 공격 분석(1)

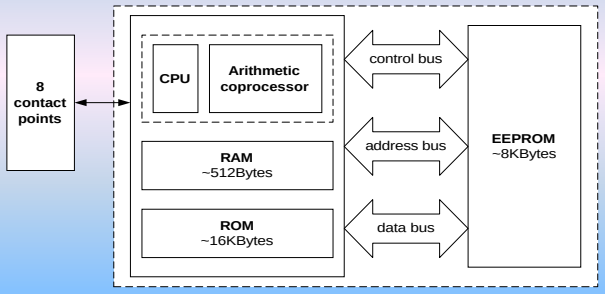
❑ SC contacts (ISO/IEC 7816 part 2)



- C1=Vcc : power supply
- C2=RST : reset
- C3=CLK : clock
- C4=N.C.
- C5=GND : ground ➔ 저항/측정 포인트
- C6=Vpp : EEPROM writing voltage ➔ EEPROM 프로그래밍
- C7=I/O : input/output ➔ EEPROM 데이터입력
- C8=N.C.

스마트카드에 대한 부채널 공격 분석(2)

✓ DPA 분석 실험을 위한 메모리 활용 방안
- EEPROM ➔ 프로그램 코드(Ex) DES, RSA (assembler)
- RAM ➔ 데이터 (Ex) 내부 레지스터



마이크로프로세서 카드의 구조

스마트카드에 대한 부채널 공격 분석(3)

❑Traditional / Actual Systems

Figure: Traditional Cryptographic Assumptions

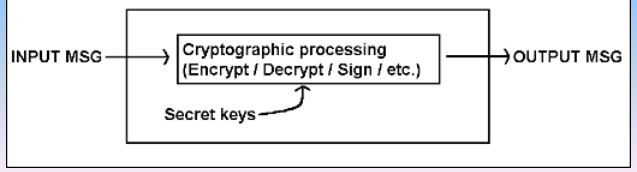
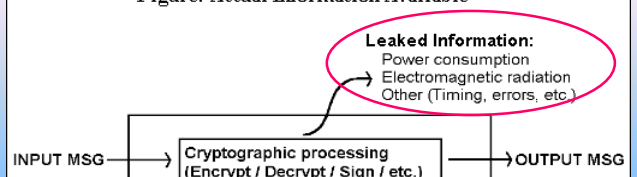


Figure: Actual Information Available



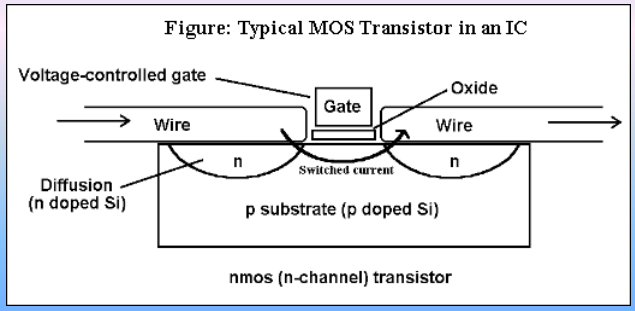
스마트카드에 대한 부채널 공격 분석(4)

❑Power Analysis

- IC 회로의 기본회로 → NMOS -FET
- Gate 통과하는 충전전류 → TR 통과

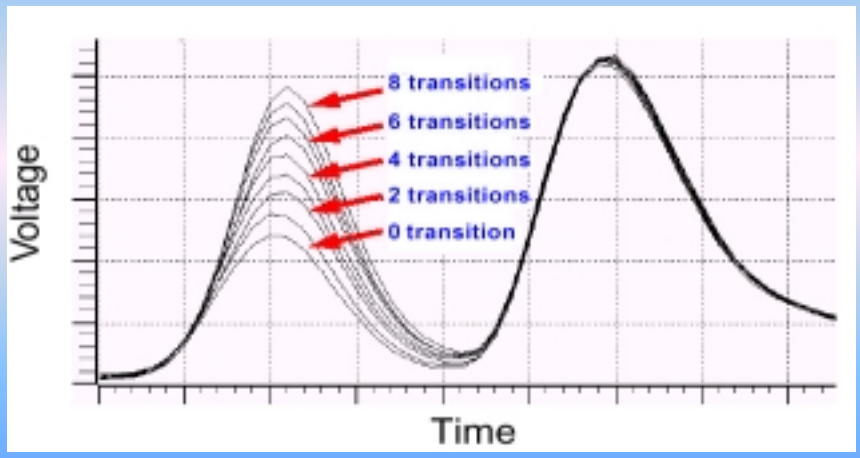
  → 전력 소모 노출, 전자기파 방사

Figure: Typical MOS Transistor in an IC



스마트카드에 대한 부채널 공격 분석(5)

- Number of Bit Transitions vs. Power Consumption
  - LDA instruction → 6.5mV difference in ith and (i+1)th transition
  - Ex) LDA A,#00h → LDA A,#FFh (8 -bit transitions) → 52 mV



스마트카드에 대한 부채널 공격 분석(6)

- 부채널 공격 요약

항목	TA	FA	PA	비고
공 격 개 념	스마트카드에 내장된 **비밀키 값에 따른** 마이크로코드상에서의 **동작시간상의 차이점을** 비교 분석하는 공격기법	**외부에서 필요에 의한 오류값을 주입**→ 마이크로코드상에서의 **외형적인 값(동작시간, 전력 정보 등)과 비교 값과의 차이점을** 분석하는 공격기법	스마트카드에 내장된 비밀키 값에 기반한 마이크로코드상에서의 전력 측정값이 여러 종류의 평문(암호문) 입력변화에 따른 **전력소모의 차이로** 나타남을 이용하는 통계적 데이터 분석 공격기법	PA공격방법이 가장 강력하다고 알려짐
전 제 조 건	- 암호시스템 제공 - 마이크로코드 제공	- 암호시스템 제공 - 마이크로코드 제공 - 알려진 키에 대한 기준 값(reference) 제공	- 암호시스템 제공 - 마이크로코드 제공 - 상당수의 평문-암호문쌍 제공	
최 초 제 안	Paul Kocher CRYPTO'96 (Cryptography Research Co., USA),	Biham & Sharmir, CRYPTO'97	Paul Kocher, Jaffe & Jun, CRYPTO'99 (98' Homepage)	

스마트카드에 대한 부채널 공격 분석(7)

❑ 전력분석 공격 요약

항목	SPA	DPA(HO -DPA)	IPA	비고
공 격 개 념	스마트카드에서 연산되는 암호 프로세서의 전력소비를 직접 관찰하여 카드 내부에 저장되어 있는 **비밀키를 직접 공격하는** 방법	스마트 카드가 암호학적 연산을 실행 시에 소비되는 전력을 표본화하여 그 데이터를 수집한 다음, **표본화된 데이터를 잡음 신호 감소와 차분 신호의** 명확성을 위해 디지털 신호 해석과 **통계적인 방법으로 분석하는** 공격 기법	공격자가 필요로 하는 정도의 소비전력 신호를 가질 수 없고 평문과 암호문의 쌍을 가질 수 없다면, 암호 시스템이 암호 알고리즘을 실행할 때 **몇 번째 키가 어디에서 반응하는지** 그 시점을 확인한 후 **키 정보를 추출하는** 공격 기법	DPA 공격방법이 가장 강력하다고 알려짐.
전 제 조 건	- 암호시스템 제공 - 마이크로코드 제공	- 암호시스템 제공 - 마이크로코드 제공 - 상당수의 평문-암호문 쌍 제공	- 암호시스템 제공 - 마이크로코드 제공 - 제한적인 평문-암호문쌍 제공	
최 초 제 안	Paul Kocher, Jaffe & Jun, '98 homepage /CRYPTO'99	Paul Kocher, Jaffe & Jun, CRYPTO'99	P. Fahn and P. Pearson, CHES'99.	

스마트카드에 대한 부채널 공격 분석(8)

❑ RSA에 대한 전력분석 공격 요약

항목	SEMD (Single-exponent, multiple data)	MESD (Multiple-exponent, single data)	ZEMD (Zero-exponent, multiple data)	비고
공 격 개 념	**이미 알고있는 키와 스마트 카드를 이용하여** 미지의 공격하고자 하는 스마트카드의 비밀 키를 알아내려고 하는 공격방법	비밀 키를 사용하여 **멱승 과정에서의 평균 소비전력을 구한 다음** 이를 이용하여 각각에 대한 **차분 데이터를 계산하여 더 긴 시간 동안 "0"이 나타나는 경우가** 올바른 키로 추정하는 공격방법	**특정 바이트의 해밍 중을 분류함수로 지정하여** 전력을 두 부류로 분류한 후 **DPA** 공격 방법 적용	ZEMD 방법이 가장 일반적인 공격법이며,세 가지 중에서 제일 적은 소비전력 신호를 필요로 함.
전 제 조 건	- RSA를 공격하기 위해서는 많은 수의 소비전력 신호가 제공되어야 한다.(예, 16비트 키를 찾기 위하여 비트당 2000개의 전력신호가 필요함) - **한 장의 키를 아는 비교용 스마트카드 + 공격대상 카드(전체 2장)**	- 공격자가 일정한 값을 자신이 선택한 지수를 사용하여 멱승과정을 수행할 수 있다. **-한 장의 키를 모르는 비교용 스마트 카드 + 공격대상카드 -(전체 2장)**	- 공격자가 일정한 값을 자신이 선택한 지수를 사용하여 멱승과정을 수행할 수 있다. -공격자는 어떤 멱지수에 대해 알 필요도 없는 대신에 오프라인 시뮬레이션을 이용하여 멱승과정에서의 중간값에 대한 결과를 예측 - **(공격대상 카드 1장)**	
제안자	T.S. Messerges, E.A. Dabbish, and R.H. Sloan, CHES'99			

스마트카드에 대한 부채널 공격 분석(9)

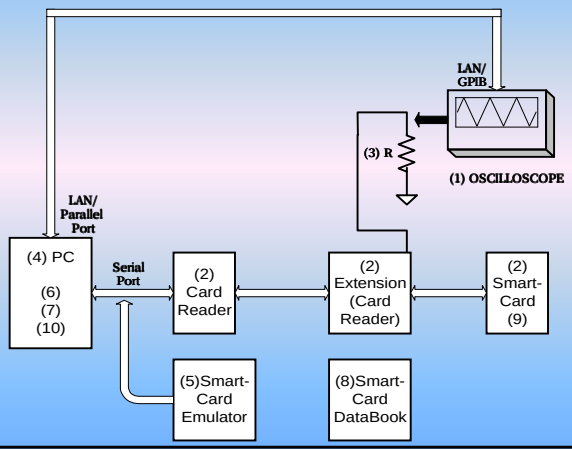
❑ 전력분석 공격 대응방안 요약

항목	SPA	DPA(HO -DPA)	IPA	비고
방어기법	❑ **조건 점프 명령**(conditional branching operation)을 **배제**시키는 방법 ❑ **하드웨어적인 구현**(hard-wired hardware implementation) 기술 적용 방법	❑ 신호 크기를 줄이는 방법 ❑ **랜덤화된 회로를 추가하는** 방법 ❑ 암호 알고리즘 설계 시 키 스케쥴링 과정의 비선형화 방법 ❑**랜덤 타이밍** 이동 기법의 적용 ❑핵심이 되는 명령을 분석이 어려운 다른 어셈블러 명령으로 대체하는 기법 ❑ 주어진 알고리즘에서 계산의 명료화 기법(explicit way of computing)을 도입하여도 획득된 알고리즘에서 DPA가 불가능하도록 하는 방법	❑ **키 비트의 여러 번 사용을 기피하는** 방법, ❑ 프로파일링 단계에서 키에 대한 위치 정보를 모르게 하기 위해서 동일한 키에 대한 연산을 한번만 하는 것이 아니라 여러 번 수행하여 키에 대한 영향을 여러 곳에 전파시키는 방법 ❑ **소스 코드의 실행을 랜덤화** 시키는 방법 ❑ 데이터 표현을 랜덤화 시키는 방법	

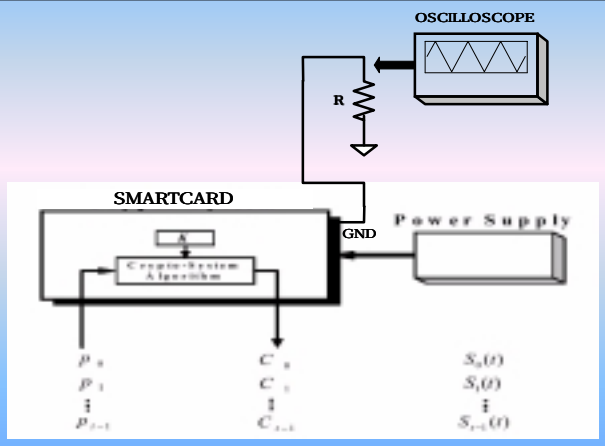
DES의 DPA 실험(1)

항 목	실험 내용
1. 실험 준비 사항	❑ (1) 디지털 오실로스코프 1대(1 GHz 이상 샘플링 기능 보유; PC 상으로 디지털 데이터의 저장이 가능할 것. ❑ (2) 스마트카드 리더기 및 카드 각 1대 (카드 리더 확장기판 포함) ❑ (3) 가변 저항 10 150 ohm 1개 ❑ (4) 퍼스널 컴퓨터 1대(데이터 저장 및 분석용; Windows95 & P-III 이상) ❑ (5) 스마트카드용 에뮬레이터 1대(선택사항) ❑ (6) 스마트카드용 어셈블러(선택사항) ❑ (7) 스마트카드 구동용 PC 프로그램(선택사항) ❑ (8) 스마트카드용 마이크로프로세서 데이터 북(선택사항) ❑ (9) 스마트카드 탑재용 어셈블리 프로그램 구현 ❑ (10)전력분석용 PC 시뮬레이션 프로그램 (분류데이터 생성용 프로그램; DPA 분석용 프로그램) 구현

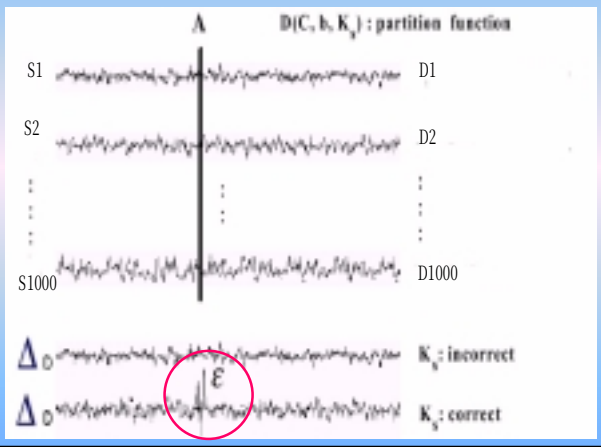
DES의 DPA 실험(2)

항 목	실험 내용
2. 실험 환경 구성 도	

DES의 DPA 실험(3)

항 목	실험 내용
3. DPA 전력측정 방법	

DES의 DPA 실험(4)

항 목	실험 내용
3. DPA 전력측정 방법(계속)	

DES의 DPA 실험(5)

항 목	실험 내용
3. DPA 전력측정 방법(계속)	1) DES 알고리즘을 어셈블리어로 구현한다. ※ 첨부 A. 어셈블리 프로그램 소스 리스트 참조할 것 2) 카드 리더기를 통하여 EEPROM 영역에 확장된 INTEL-16진 형태(extended INTEL-hexadecimal format)의 명령어 코드를 write한다.(대략적으로 코딩 사이즈는 4kbyte 내외임) 3) 카드 리더기와 측정 장치를 구성한다. 4) N=0에 대한 첫 번째 평문 데이터를 입력한 후 프로그램을 실행한다. 실행 결과 분류 함수값과 암호문 데이터를 저장한다. 그리고 오실로스코프를 이용하여 전력 파형을 측정하여 데이터 파일로 저장한다.

DES의 DPA 실험(6)

항 목	실험 내용
3. DPA 전력측정 방법(계속)	5) N=1,2,...,999에 대하여 4)의 과정을 반복한다. 6) N=1,000개의 분류함수 계산 결과 및 평문-암호문 데이터를 파일로 저장한다. 평문 데이터 $P[0], P[1], P[2], \cdots, P[N-1] \quad (N=1000)$ (Hexa-decimal) 암호문 데이터 $C[0], C[1], C[2], \cdots, C[N-1] \quad (N=1000)$ (Hexa-decimal) 전력 데이터 $S[0][0], S[0][1], \quad S[0][2], \cdots, S[N-1][M-1] \quad (N=1000, M=100000)$ 분류함수 $D[i] = D(C_1[i], C_6[i], K_6[1]) = C_1[i] \oplus SBOX1(C_6[i] \oplus K_6[1])$ 7) N=1,000 종류의 전력 소모 데이터로부터 S-box #1이 작용하는 시점에서의 데이터 값을 예측한 후 상기 분류 함수에 따라 다음과 같이 전력 신호를 두 개의 그룹으로 나눈다. $S_0 = \{S[i][j] \mid D(\cdot,\cdot,\cdot) = 0\}$ $S_1 = \{S[i][j] \mid D(\cdot,\cdot,\cdot) = 1\}$

DES의 DPA 실험(7)

항 목	실험 내용
3. DPA 전력측정 방법(계속)	8) 각 세트에 대하여 평균 전력 신호를 계산한다. $A_0[j] = \frac{1}{\lvert S_0 \rvert} \sum S[i][j] \quad if \quad S[i][j] \in S_0$ $A_1[j] = \frac{1}{\lvert S_1 \rvert} \sum S[i][j] \quad if \quad S[i][j] \in S_1$ 9) 두 개의 평균값을 서로 뺀 이산 시간 DPA 바이어스 신호를 구한다. $T[j] = A_1[j] - A_0[j]$ 10) 다음과 같이 기대값으로 진짜 키(correct key)인지 여부를 결정한다. $T[j] = E\{S[i][j] \mid D(\cdot,\cdot,\cdot) = 0\} - E\{S[i][j] \mid D(\cdot,\cdot,\cdot) = 1\}$ $= E\{S[i][j]\} - E\{S[i][j]\} = A_0[j] - A_1[j] = 0 \quad \forall j$ (incorrect key) $T[j] = E\{S[i][j] \mid D(\cdot,\cdot,\cdot) = 0\} - E\{S[i][j] \mid D(\cdot,\cdot,\cdot) = 1\} \neq 0$ (correct key)

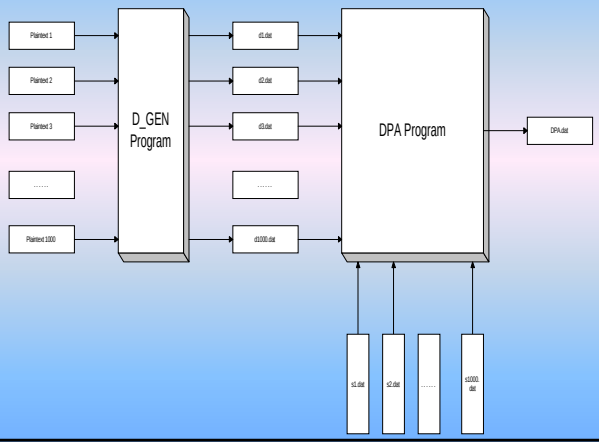
DES의 DPA 실험(8)

항 목	실험 내용
3. DPA 전력측정 방법(계속)	11) 진짜 키가 아니면 6-비트 키 값을 다르게 설정하여 진짜 키를 찾을 때까지 최대 2^6=64번까지 7) 10)과정을 반복한다. 12) 나머지 7개의 S-box #2 #8에 대하여 새로운 키 그룹에서 6-비트의 키 값을 추정하여 7) 10)과정을 실행한다.(본 단계가 끝나면 6*8=48비트의 보조키를 모두 찾을 수 있게 된다.) 13) 56-비트 키 중에서 찾아지지 않은 나머지 8-비트는 기지 평문 공격을 이용하여 평문-암호문 쌍을 이용한 공격으로 찾는다. * 공격 복잡도 = 2^6 * 8 * 2^8 = 2^17 임.

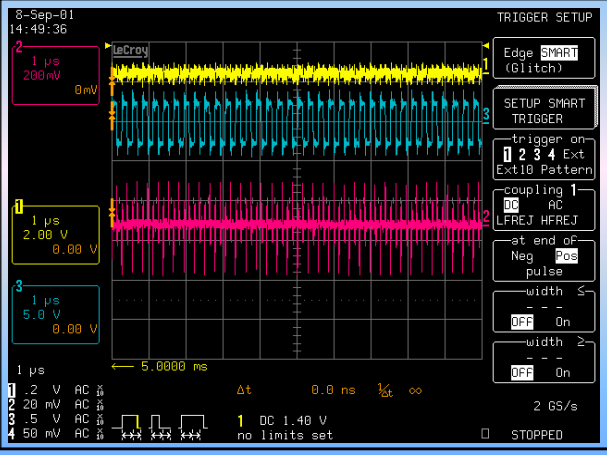
DES의 DPA 실험(9)

항 목	실험 내용
4. DPA 분석용 프로그램 구현	1) 공격 대상 암호 알고리즘 : DES (또는 3-DES :triple DES) 2) 카드 리더기 분석 대상 시스템 - 카드 리더기 - 스마트카드: 8-비트 CPU가 내장된 스마트카드 3) 카드 에뮬레이터 분석 대상 시스템 - 카드 에뮬레이터 : In-Circuit Emulator - 스마트카드 프로브 : 스마트카드 대용 4) 암호 시스템 구현: 스마트카드전용 CPU 어셈블러 5) DPA 시뮬레이션 프로그래밍 : VC++6.0

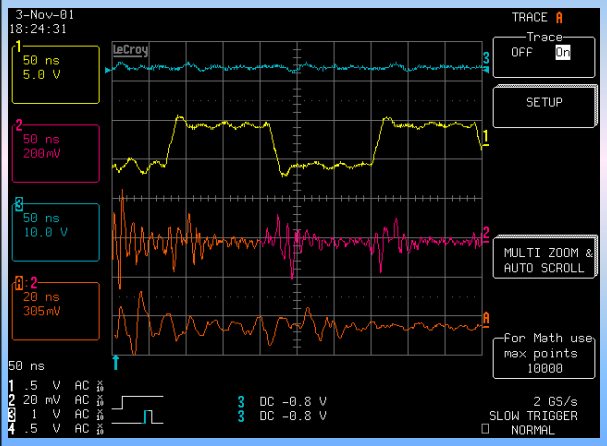
DES의 DPA 실험(10)

항 목	실험 내용
4. DPA 분석용 프로그램 구현 (계속)	

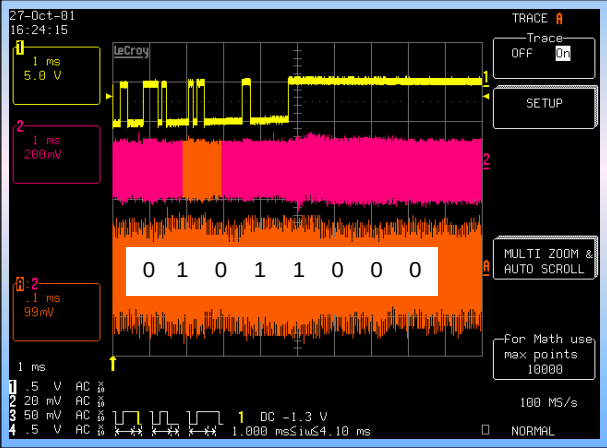
DES의 DPA 실험(11)

항 목	실험 내용
5. 전력 파형 측정 - 클럭에 따른 전력 파형 (트리거신호, 클럭, 전력 파형) → "클럭 edge에서 전력소모가 크게 나타남"	

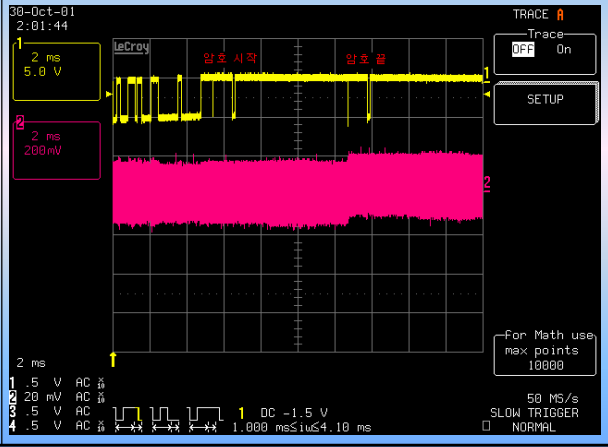
DES의 DPA 실험(12)

항 목	실험 내용
5. 전력 파형 측정 (계속) - 클럭에 따른 전력 파형 (트리거신호, 클럭, 전력 파형, 전력파형 확대) → "(확대) 클럭 edge에서 전력소모가 크게 나타남"	

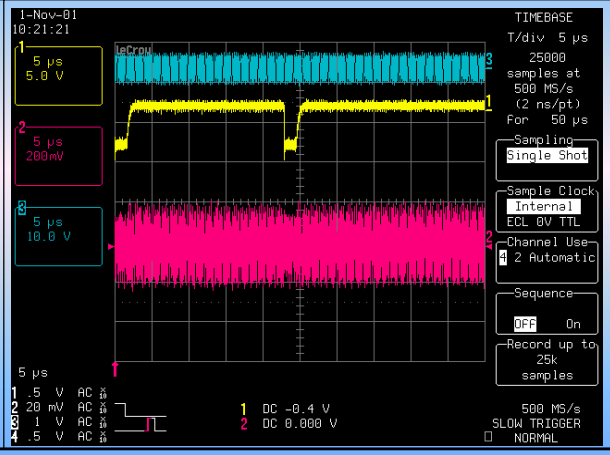
DES의 DPA 실험(13)

항 목	실험 내용
5. 전력 파형 측정 (계속) - Serial data 에 따른 전력 파형 (Serial data 신호, 전력 파형, 전력파형 확대) → 데이터 "1"일 때의 전력이 "0"일 때의 전력소모보다 크게 나타남 → 전력소모량으로 구분 가능	

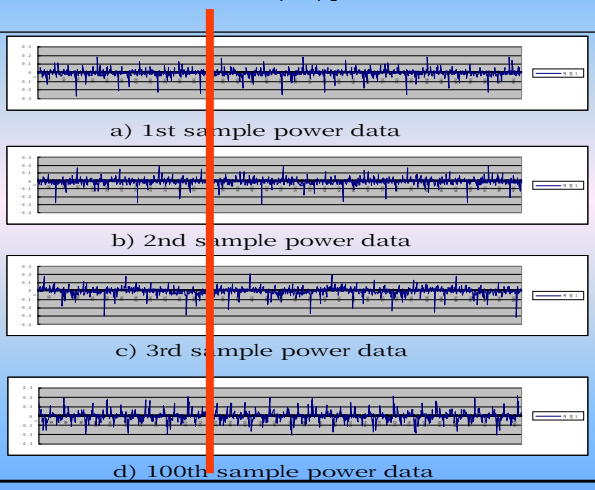
DES의 DPA 실험(14)

항 목	실험 내용
5. 전력 파형 측정 (계속) - 카드 리더기 파형 - Serial data 핀으로 시작신호 /끝신호 출력 파형 (Serial data 신호, 전력 파형) ➔ 전력데이터 수집 시점 결정 (DES 작동 시간 = 8ms)	

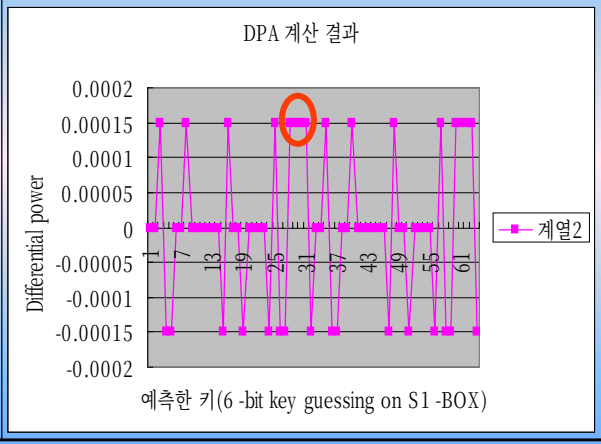
DES의 DPA 실험(15)

항 목	실험 내용
5. 전력 파형 측정 (계속) - 에뮬레이터 파형 - DES의 S8 -box에서의 시작과 끝을 나타내는 파형 (Serial data 신호, 트리거 신호, 전력 파형) ➔ 전력데이터 수집 시점 결정 (S -box1 작동시간=20us)	

DES의 DPA 실험(16)

항 목	실험 내용
5. 전력 파형 측정 (계속) - 수집된 그룹 1 전력 샘플데이터(DES의 S1-box 시작 위치에서의 1,000 샘플포인트, 카드 리더기 활용) - 그룹 1 전력 샘플 : 100개 - 그룹 2 전력 샘플 : 100개 - 그룹 3 전력 샘플 : 100개	

DES의 DPA 실험(17)

항 목	실험 내용
5. 전력 파형 측정 (계속) - s1-box에 대한 결과 (correct key=011100B=28D 가정, group 1) ➔ 유사 키 군을 추정함 ➔ 정확한 키의 추정은 1000개 이상의 정밀한 전력 데이터 수집이 요구됨	

DES의 DPA 실험(18)

완전한 암호분석을 위한 조건

❑ 1) **스마트카드에 대한 사전 정보가 충분하게 제공되어야** 한다. 암호 분석을 위하여 마이크로프로세서 구조(또는 하드웨어 구조), 메모리 구조, 활용 메모리(RAM, PROM, EEPROM 등) 등에 대한 사전 정보가 충분하게 제공되어야 할 필요가 있으며, 이는 4장에 언급될 분석 사례에서 알 수 있는 바와 같이 스마트카드 **개발 회사에서** 자체분석이 이루어지거나 **또는 개발회사와 공동으로** 분석이 이루어질 필요성을 보여주고 있다.

❑ 2) **스마트카드에 대한 응용 프로그램, 암호화 프로그램(S/W 또는 H/W)을 탑재하거나 또는 적절히 수정·변경할 수 있는 환경이 갖추어질 필요가** 있다. 스마트카드를 PC상에서 구동시키거나 또는 암호화 알고리즘의 마이크로코드를 분석하여 예측 대상인 부분 키가 작동하는 위치를 정확하게 추정하지 않으면 실험분석은 실패할 가능성이 높다.

❑ 3) 경우에 따라서는 **스마트카드용 마이크로프로세서에 대한 에뮬레이터를 필요로** 할 수도 있다. (마이크로코드에 대한 분석이나 특정 명령어의 정확한 동작 타이밍 정보 등을 분석)

❑ 4) **아주 정밀한 수준의 계측 장비를 보유하여야** 한다. 수~수십 Gbps의 샘플링이 가능하고, 디지털 데이터 저장능력(예, 수십 ms 분량의 데이터 저장, 수백 Mbyte급)이 높은 계측장비가 필요하며, 컴퓨터 시뮬레이션을 위한 수집 데이터 저장을 위하여 PC와 연동될 필요가 있다.

❑ 5) **수집된 모든 종류(예, 1,000종)의 데이터 시작 위치를 거의 정확히 일치시킬 필요성이 있으며, 이에 따른 계측기의 정확한 트리거 기능이** 요구된다. 그렇지 않을 경우 키 작용 시점에 대한 정보가 분산될 수 있기 때문에 데이터의 신빙성이 떨어질 수 있다.

DES에 대한 DPA 방어기술 제안(1)

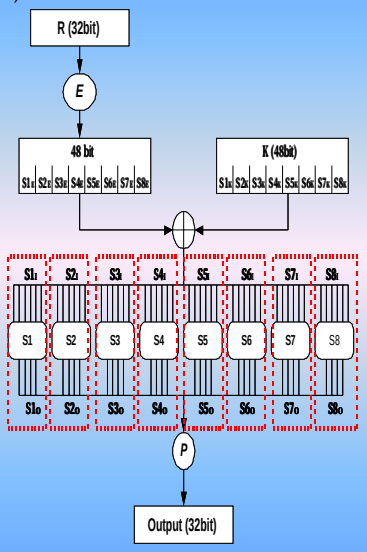
항 목	1) 기존 방안 (6-bit key guessing)	2) 두 S-box의 혼합 구현 방안 (12-bit key guessing)	3) 네 S-box의 혼합 구현 방안 (24-bit key guessing)	4) 여덟 S-box의 혼합 구현 방안 (48-bit key guessing)
제 안 내 용	원래의 방법으로 S-box를 순차적으로 마이크로코드화 시킬 경우이므로 별도의 대책 강구	인근하는 S -box를 두 개씩 묶어서 마이크로코드화 시킬 경우임.	인근하는 S -box를 네 개씩 묶어서 마이크로코드화 시킬 경우임.	S-box를 모두 묶어서 마이크로코드화 또는 하드웨어 구현할 경우임.
S-box 입출력비트수	Input : 6-bit Output: 4-bit # of S-boxes: 8	Input : 12-bit Output: 8-bit # of S-boxes: 4	Input : 24-bit Output: 16-bit # of S-boxes: 2	Input : 48-bit Output: 32-bit # of S-boxes: 1 (Input : 6-bit Output: 4-bit # of S-boxes: 8 parallel clocked)

DES에 대한 DPA 방어기술 제안(2)

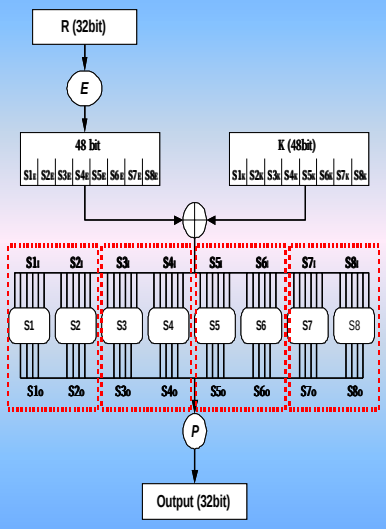
항 목	1) 기존 방안 (6-bit key guessing)	2) 두 S-box의 혼합 구현 방안 (12-bit key guessing)	3) 네 S-box의 혼합 구현 방안 (24-bit key guessing)	4) 여덟 S-box의 혼합 구현 방안 (48-bit key guessing)
공 격 순 서	S1-box → S2-box → S3-box → S4-box → S5-box → S6-box → S7-box → S8-box	S1, S2 -box ↓ S3, S4-box ↓ S5, S6-box ↓ S7, S8-box	S1, S2, S3, S4 ↓ S5, S6, S7, S8	S1, S2, S3, S4, S5, S6, S7, S8
단계 수	8	4	2	1
복잡 도	2^6 x 8 x 2^(56-48) = 2^17	2^12 x 4 x 2^(56-48) = 2^22	2^24 x 2 x 2^(56-48) = 2^33	2^48 x 1 x 2^(56-48) = 2^56

DES에 대한 DPA 방어기술 제안(3)

(1) 기존 방안 ➔ 2^17

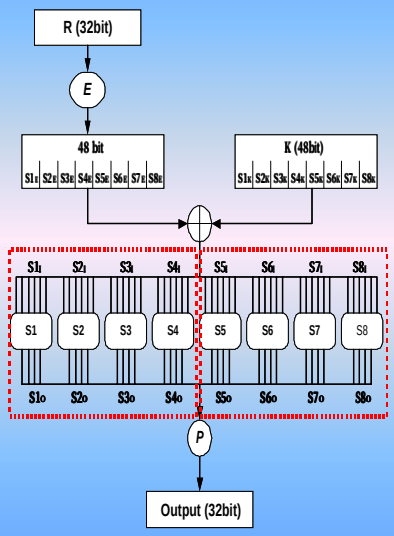


(2) 대응 방안 -1 ➔ 2^22

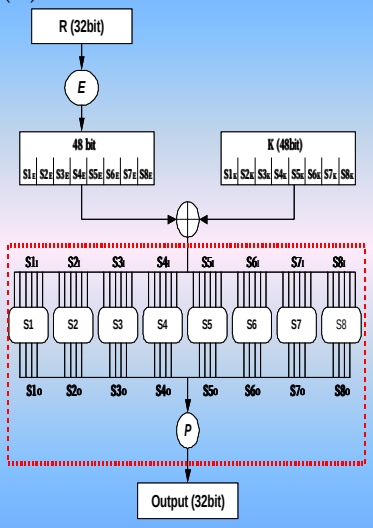


DES에 대한 DPA 방어기술 제안(4)

(3) 대응 방안 -2 ➔ 2^33



(4) 대응 방안 -2 ➔ 2^56



결 론

- SCSUG-SCPP, FIPS 140-2 CMVP 분석
  - TA/FA/PA 관련항목 분석
- 부채널 공격 기법 비교 분석
  - TA/FA/PA
  - SPA/DPA/IPA/HO -DPA
  - SEMD/MESD/ZEMD: RSA
- DPA 공격 실험 ➔ DES
  - 8 -bit CPU 스마트카드
  - 전력 데이터 파형100개씩 축소모델 ➔ 3개 그룹 실험
  - 카드 리더기, 카드 에뮬레이터
  - 유사 키 군에 대한 추정 (정밀 키의 추정은 많은 데이터 요구)
- 방어대책 제안
  - S-box를 기존1개씩/2개씩/4개씩/8개씩 그룹화 구현 (병렬 프로세싱)
  - 비도 수준 : 2^17 ➔ 2^22 ➔ 2^33 ➔ 2^56(하드웨어)
- 향후 연구 방향
  - DES 정밀 실험
  - RSA/ECC에 대한 실험/대응방안 연구

References

1. 박창섭, ***암호이론과 보안***, 대영사, 1999
2. 최용락외, ***컴퓨터 통신 보안***, 도서출판 그린, 2001
3. William Stalling, ***Cryptography and Network Security (2nd Ed.)***, Prentice -Hall, Inc., 1999
4. B. Schneier, ***Applid Cryptography (2nd Ed.),*** John Wiley & Sons, Inc., 1995.
5. A. Menezes, ***Handbook of Applied Cryptography***, CRC Press, 1997.
6. D. Stinson, ***Cryptography – Theory and Practice (2nd Ed.),*** CRC Press, 2002.