차분 전력 분석에 안전한 역원기의 설계

DPA-Resistant Design of the Inverter

김 희 석, 조 영 인, 한 동 국, 홍 석 희*
(HeeSeok Kim, Young In Cho, Dong-Guk Han, SeokHie Hong*)

Abstract : In the block cipher, DPA-resistant masking methods make an appropriation of extremely high cost for the non-linear part. Block ciphers like AES and ARIA use the inversion operation as this non-linear part. This make various countermeasures be proposed for reducing the cost of masking inversion. In this paper, we propose the efficient masking inverter by rearranging the masking inversion operation over the composite field and finding duplicated multiplications.

Keywords: differential power analysis, masking method, inverter, AES

I. 서론

수학적으로 안전한 것으로 알려진 알고리즘조차도 구현 단계에서 고려되지 못한 부가적인 정보의 누출이 있다는 것이 알려졌고, 이로부터 비밀 키의 값을 알아낼 수 있는 부채널 공격(Side Channel Attack)이 소개되었다[12]. 이러한 부채널 공격이 소개되면서 많은 암호시스템 설계자들은 효율적인 대응법들을 연구하기 시작했고, 부채널 공격 중 하나인 차분 전력 분석(Differential Power Analysis, DPA)[10,11,13]에 대한 대응법으로는 마스킹 대응법(masking method)이 활발히 연구되어지고 있다[3,5,6,7].

마스킹 기법은 평문 x에 대하여 암호문 y를 얻기 위해 마스킹 난수 m을 이용 $x\oplus m$(⊕: xor)의 암호문 $y'(=y\oplus m')$ 을 구한 후, 최종적으로 y를 얻기 위해 $y'\oplus m'$의 연산을 수행한다.(경우에 따라 마스킹 기법은 다르게 구성한다.) 따라서 암호화 중 중간 값을 알 수 없기 때문에 일반적인 전력 분석 공격은 성공할 수 없다. 이러한 마스킹 기법을 사용한 경우, $x\oplus m$의 암호문 $y'(=y\oplus m')$에서 m'을 알아야 실제 원하는 암호문 y를 얻을 수 있다. 하지만 블록암호 알고리즘은 비선형 연산을 수행하므로 수정되지 않은 블록 암호 시스템에서 m' 값은 x에 따라 다르며 이 값을 중간 값의 누출 없이 아는 것도 상당한 연산을 필요로 한다.

AES, ARIA와 같은 블록 암호에서 이러한 비선형 연산은 역원 연산에 의해 수행된다. 본 논문에서는 덧셈 마스킹(additional masking)[2,6]을 사용하여 일차 차분 전력 분석에 안전한 역원기를 설계한다. 즉, $x\oplus m$으로부터 중간 값의 누출 없이 $x^{-1}\oplus m'$을 연산하는 마스킹 역원 계산 방법을 제안한다. 마스킹 역원 계산 방법의 기존 방법들은 그 효율성을 위해 복합체(composite field) 위에서의 연산[4]을 사용한다. 본 논문도 그 효율성을 위해 복합체 위에서의 마스킹 방법을 고려하였고, 새로운 연산 식을 세워 기존 방법들에 비해 보다 효율적인 연산 방법을 제시하였다.

본 논문의 구성은 다음과 같다. 2절은 AES S-box에 대해 설명하고, 3절에서는 제안하는 마스킹 역원 계산 방법에 대해 소개한다. 본 논문에서 제안하는 마스킹 방법의 안전성은 4절에서 소개한다.

* 책임저자(Corresponding Author)
논문접수 : 2008. 8. 12..
김희석, 조영인, 홍석희 : 고려대학교 정보경영공학전문대학원.
(heeseokkim@cist.korea.ac.kr, elowey@korea.ac.kr, hsh@cist.korea.ac.kr)
한동국 : 한국전자통신연구원
(christa @etri.re.kr)
※ "본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (IITA-2008-(C1090-0801-0025))

II. AES 의 역원 연산

블록 암호 알고리즘 AES(Advanced Encryption Standard)는 2001년 NIST(National Institute of Standards and Technology. 미국 국립 표준 기술원)에 의해 차세대 표준 암호 알고리즘으로 채택되었다[1]. AES의 한 라운드는 AddRoundKey, Subbytes, ShiftRows, MixColumn의 네 단계로 구성되어 있으며, 이 네 단계 중 Subbyte 연산, 즉 S-Box 연산 부분은 블록 암호 알고리즘의 비선형 연산을 수행한다. AES의 S-Box는 x^{-1}의 아핀 변환(affine transform) 형태로 다음의 연산을 수행한다.

$$S:\ GF(2^8) \rightarrow GF(2^8)$$
$$S(x) = Bx^{-1} \oplus b\,.$$

위의 연산에서 x^{-1}의 연산은 유한체 $GF(2^8)$(기약 다항식 : $x^8+x^4+x^3+x+1$) 상에서 이루어지며 이 연산을 위한 행렬과 벡터의 값은 다음과 같다.

$$B=\begin{pmatrix}1&0&0&0&1&1&1&1\\1&1&0&0&0&1&1&1\\1&1&1&0&0&0&1&1\\1&1&1&1&0&0&0&1\\1&1&1&1&1&0&0&0\\0&1&1&1&1&1&0&0\\0&0&1&1&1&1&1&0\\0&0&0&1&1&1&1&1\end{pmatrix} \qquad b=\begin{pmatrix}1\\1\\0\\0\\0\\1\\1\\0\end{pmatrix}$$

$$CD=\begin{pmatrix}0&1&0&1&0&1&1&1\\0&0&1&1&1&1&1&1\\1&1&1&0&1&1&0&1\\1&1&0&0&0&0&1&1\\0&1&0&0&0&0&1&1\\1&1&0&0&1&1&1&0\\0&1&1&0&0&0&1&1\\1&1&1&1&0&1&1&0\end{pmatrix} \qquad c=\begin{pmatrix}0\\1\\0\\0\\0\\1\\1\\1\end{pmatrix}.$$

1. 복합체(Composite Field) 위에서의 역원 연산

일반적으로 AES의 하드웨어 설계에 있어서 S-Box를 저장하고 호출하는 방식은 공간적인 제약이 많이 따르는 편이며, 이로 인해 S-Box를 연산하는 방법이 주로 사용된다. 하지만 x^{-1}의 연산과 아핀 변환으로 이루어지는 S-Box 연산은 x^{-1}, 즉 $GF(2^8)$에서의 역원계산에서 상당한 비용이 요구되며 실제 역원 계산에 들어가는 비용은 AES 라운드 연산에서 상당한 부분을 차지한다. 따라서 역원 계산의 효율성은 전체 암호 알

고리즘의 성능에 크게 영향을 미치며, 이러한 이유로 역원 계산의 비용을 감소시키기 위한 다양한 방법들이 연구되어 졌고 복합체의 개념도 소개되어졌다[4]. 즉, $GF(2^8)$위에서의 일반적인 역원계산이 아닌 역원 계산의 비용이 작은 복합체 $GF(((2^2)^2)^2)$로의 변환을 수행한 후 역원 계산을 수행, 결과 값을 다시 $GF(2^8)$위의 원소로 역변환 하는 방법이 소개되어졌다.

복합체에서의 역원연산은 부분체(subfield)에서의 연산을 통해 이루어진다. S-box 연산에서 사용되어지는 $GF(2^8)$위에서의 연산은 복합체인 $GF(((2^2)^2)^2)$ 위에서의 연산으로 변형되어질 수 있으며, 각 부분체에서 사용되어지는 기약다항식의 형태는 다음과 같다.

$GF(2^2)$ over $GF(2)$: $P_0(x)=x^2+x+1$

$GF((2^2)^2)$ over $GF(2^2)$: $P_1(x)=x^2+x+\Phi$

$GF(((2^2)^2)^2)$ over $GF((2^2)^2)$: $P_2(x)=x^2+x+\lambda$

$P_0(x)$의 근을 α, $P_1(x)$의 근을 β, $P_2(x)$의 근을 γ라 한다면 ($\alpha \in GF(2^2)$, $\beta \in GF((2^2)^2)$, $\gamma \in GF(((2^2)^2)^2)$), $GF(2^2)$ 의 모든 원소는 $a_1\alpha+a_0$, $GF((2^2)^2)$의 모든 원소는 $(a_3\alpha+a_2)\beta+a_1\alpha+a_0$, $\gamma \in GF(((2^2)^2)^2)$의 모든 원소는 $((a_7\alpha+a_6)\beta+a_5\alpha+a_4)\gamma+(a_3\alpha+a_2)\beta+a_1\alpha+a_0$의 형태로 표현된다. 연산의 효율성을 위해 $P_1(x)$, $P_2(x)$가 기약인 성질을 만족하는 원소 중에 $GF(2^2)$의 원소 Φ는 $\alpha((10)_2)$로, $GF((2^2)^2)$의 원소 λ는 $(\alpha+1)\beta((1100)_2)$로 선택하였다.

복합체 위에서의 역원 연산은 다음의 연산을 통해 이루어진다. $A \in GF(((2^2)^2)^2)$의 역원 A^{-1} 연산을 위해 $C^{-1}A^{16}$ ($C=A^{17} \in GF((2^2)^2)$)을 연산한다. 또한 $C \in GF((2^2)^2)$의 역원 C^{-1}는 $D^{-1}C^4$ ($D=C^5 \in GF(2^2)$) 연산을 통해 이루어진다. 즉 $GF(((2^2)^2)^2)$위에서의 역원 연산이 $GF(2^2)$위에서의 역원 연산을 통해 이루어진다. 물론, 역원 연산을 위해 추가적인 A^{16}, A^{17}과 같은 연산이 필요하지만 복합체 연산의 특성상 $A=a_h\gamma+a_l$의 16승 연산 결과는 $a_h\gamma+(a_h+a_l)$의 4비트 xor 연산, $(a_h\gamma+a_l)^{17}$의 연산 결과는 $a_h^2\lambda+(a_h+a_l)a_l$로 $GF((2^2)^2)$위에서의 한 번의 제곱 연산과 곱셈 연산만을 요구한다. 다음 그림은 $GF(((2^2)^2)^2)$위에서의 역원 연산과 $GF((2^2)^2)$위에서의 곱셈, 제곱 연산을 도식화한 것이다.

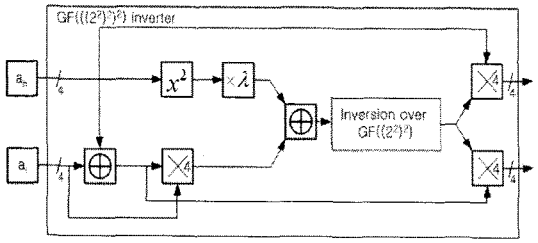


그림 1 $GF(((2^2)^2)^2)$ 역원 연산

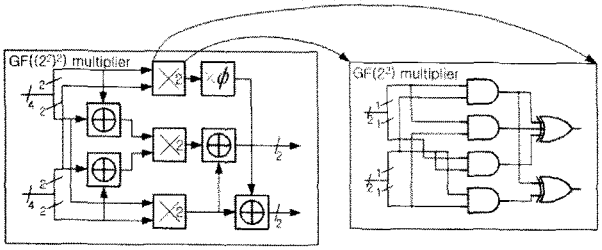


그림 2 $GF((2^2)^2)$ 곱셈 연산

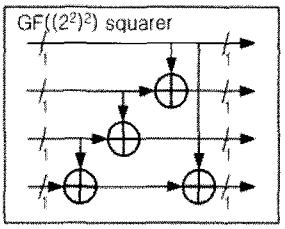


그림 3 $GF((2^2)^2)$ 제곱 연산

그림 1에 따른 $A=a_h\gamma+a_l$의 역원 $A^{-1}=a_h'\gamma+a_l'$를 계산하는 수식은 식 (1)과 같다.

$$\begin{aligned} d &= a_h^2\lambda \oplus a_l(a_h \oplus a_l) \\ d' &= d^{-1} \\ a_h' &= a_h d' \\ a_l' &= (a_h \oplus a_l)d' \end{aligned} \qquad (1)$$

III. 제안하는 역원 마스킹

마스킹 역원 계산의 입력 마스킹을 $m=(m_h \| m_l)$, 출력마스킹을 $m'=(m_h' \| m_l')$이라 정의했을 때, 제안하는 마스킹 역원 계산은 다음의 네 단계로 이루어진다.

1 단계. $A+m=(a_h \oplus m_h \| a_l \oplus m_l)$로부터 $d \oplus m_d$의 계산.

2 단계. $d \oplus m_d$로부터 $d' \oplus m_d'$의 계산

3 단계. $d' \oplus m_d'$과 $a_h \oplus m_h$ 로부터 $d'a_h \oplus m_h'$의 계산

4 단계. $d' \oplus m_d'$, $a_h \oplus m_h$, a_l+m_l로부터 $d'(a_h \oplus a_l) \oplus m_l'$의 계산

각 단계의 연산에서 나타날 수 있는 모든 중간 값은 A의 8비트 값과 특정 상수로부터 생성되어질 수 있는 모든 중간 값에 독립적이어야만 한다. m_d, m_d'은 $GF((2^2)^2)$의 마스킹 역원 계산에 대한 입출력 마스킹 값으로 본 논문에서는 효율적인 연산을 위해 $m_d'=m_h$로 선택한다.

본 논문에서는 2 단계의 $GF((2^2)^2)$위에서의 마스킹 역원 계산은 언급하지 않는다. 이는 $GF(((2^2)^2)^2)$ 위에서의 마스킹 역원 방법이 $GF((2^2)^2)$에서도 동일하게 적용될 수 있기 때문이다. $GF(((2^2)^2)^2)$에서의 역원 연산은 그림 2, 그림 3에서 보여지는 것처럼 $GF((2^2)^2)$ 곱셈기에 대부분의 비용이 소요된다. 따라서, 본 논문에서는 마스킹 역원 계산에서 사용되어지는 $GF((2^2)^2)$ 곱셈기의 개수를 최소화하는 방법을 고려하였다. 쉬운 기술을 위해 다음과 같은 기호를 정의한다.

- $\tilde{a}_h = a_h \oplus m_h$
- $\tilde{a}_l = a_l \oplus m_l$
- $\tilde{d}' = d' \oplus m_h$
- $\tilde{a}_h' = a_h d^{-1} \oplus m_h'$
- $\tilde{a}_l' = (a_h \oplus a_l)d^{-1} \oplus m_l'$

1 단계. $d \oplus m_d$의 연산

$d \oplus m_d(x_h^2\lambda \oplus a_l(a_h \oplus a_l) \oplus m_d)$는 다음의 수식 연산을 통해 이루어진다.

$$\underbrace{\lambda \tilde{a}_h^{\,2}}_{a} \oplus \underbrace{m_h(\tilde{a}_h \oplus \tilde{a}_l)}_{M1} \oplus m_d \oplus \underbrace{\tilde{a}_l(\tilde{a}_h \oplus \tilde{a}_l)}_{b}$$
$$\oplus \underbrace{m_h m_l}_{M2} \oplus \tilde{a}_h \underbrace{(m_h \oplus m_l) \oplus m_l^{\,2}}_{X3} \oplus \lambda \underbrace{m_h^{\,2}}_{S1} \quad (2)$$

각 곱셈, 제곱, 상수곱 연산에 대한 안전성은 다음 절에 논의한다. 식 (2)의 각 곱셈, 제곱, 상수곱에 대한 결과 값들을 덧셈 연산(XOR 연산)할 때에는 의미있는 중간 값과 연관성이 없도록 순서를 잘 고려하여 XOR 연산하여야 한다. 예를 들어, (2)의 식을 연산하는 과정에서 a+$S1$의 연산이 먼저 수행된다면 이는 $a_h^2\lambda$값으로 전력 분석 공격에 취약점이 드러난다. 이러한 단순한 예에서 뿐만 아니라 마스킹 값의 각 비트 값이 0.5가 아닌 다른 확률로 '0' 또는 '1' 에 편중된다면 이 부분도 역시 취약점이 될 수 있다. 이러한 편중을 막기 위해 연관성이 생기는 경우는 fresh_mask(fm)를 사용하여 이를 제거해야 한다. Fresh_mask란 최종 결과의 마스킹 값과는 연관이 없는 값으로 중간 연산 결과의 데이터 편중을 막기 위해 사용되어지는 값이다. Fresh_mask 값 fm을 사용해서 식 (2)의 순서를 고려 변형하면 식 (3)과 같다.

$$(((fm \oplus a) \oplus M1) \oplus ((m_d \oplus b) \oplus M2))$$
$$\oplus((fm \oplus \lambda S1) \oplus X3) \quad (3)$$

그림 4는 식 **1** 단계에 대한 식 (3)을 도식화한 것이다.

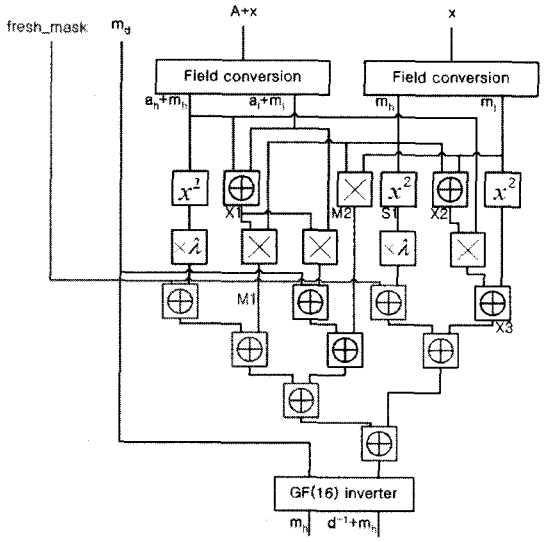


그림 4 $d \oplus m_d$의 연산

3 단계. $d'a_h \oplus m_h'$의 연산

$d'a_h \oplus m_h'$는 다음의 수식 연산을 통해 이루어진다.

$$m_h' \oplus X3 \oplus \underbrace{(\tilde{a}_h \oplus (m_h \oplus m_l))(\tilde{d}' \oplus m_l)}_{M4}$$
$$\oplus m_l' \oplus \underbrace{\tilde{d}' m_l \oplus m_l' \oplus S1 \oplus M2}_{X4} \quad (4)$$

식 (4)에서 m_l'을 두 번 더하는 이유는 fm를 사용하는 것과 같은 이유이다. 즉, 각 덧셈 연산의 결과가 m_l'으로 인해 중간 값과 완벽하게 연관성이 없는 데이터가 되기 위함이다. 식 (4)를 우선 순위를 고려하여 변형하면 식 (5)와 같다.

$$(((m_h' \oplus X3) \oplus M4) \oplus m_l')$$
$$\oplus((\tilde{d}' m_l \oplus m_l') \oplus (S1 \oplus M2)) \quad (5)$$

4 단계. $d'(a_h \oplus a_l) \oplus m_l'$의 연산

$d'(a_h \oplus a_l) \oplus m_l'$는 다음의 수식 연산을 통해 이루어진다.

$$(\tilde{a}_h \oplus \tilde{a}_l \oplus m_h)\tilde{d}' \oplus M1 \oplus X4 \quad (6)$$

식 (3)과 마찬가지로 안전성을 고려하여 순서를 재배치 하면 식 (6)은 식 (7)과 같이 변형된다.

$$((\tilde{a}_h \oplus \tilde{a}_l \oplus m_h)\tilde{d}' \oplus M1) \oplus X4 \quad (7)$$

3 단계와 4 단계의 식 (5), 식 (7)을 도식화하면 그림 5와 같다.

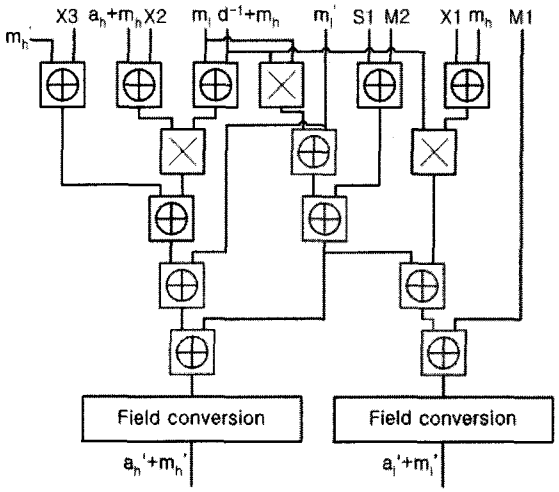


그림 5 $d'a_h \oplus m_h'$, $d'(a_h \oplus a_l) \oplus m_l'$의 연산

본 논문에서 제안하는 방법에서 fresh_mask인 fm과 GF(16) inverter의 입력 마스킹인 m_d는 새로 생성한 4비트 난수일 수도 있지만, 난수 생성에 대한 비용을 감소하기 위해 m_h' 또는 m_l'을 선택해서 사용할 수 있다. 이 값은 (3)의 식을 연산할 때 다른 값들과는 연관성이 없는 난수로 안전성에는 영향을 미치지 않는다. 하지만 fm=m_d=(m_h' or m_l')의 경우 ((fm+a)+$M1$)+ ((m_d+b)+$M2$)의 연산은 (a+$M1$+b+$M2$)을 수행하므로 $fm \neq m_d$의 값을 사용하는 것이 안전성을 더 보장할 수 있다.

IV. 제안하는 역원 마스킹의 안전성

제안하는 마스킹 방법은 다음의 Lemma 에 의해 그 안전성을 검증받을 수 있다. 다음 Lemma의 증명은 [6]의 논문을 참고한다.

Lemma 1) 원소 a가 $GF(2^n)$의 원소이고, m_a가 a와 무관한 $GF(2^n)$ 위에서 균등한 분포에 의해 선택되어진 값일 때, a+m_a는 a에 독립이다.

Lemma 2) 원소 a, b가 $GF(2^n)$의 원소이고, m_a, m_b가 a, b와 무관한 $GF(2^n)$위에서 균등한 분포에 의해 선택되어진 값일 때, $(a \oplus m_a)(b \oplus m_b)$는 a, b에 독립이다.

Lemma 3) 원소 a가 $GF(2^n)$의 원소이고, m_a, m_b가 a와 무관한 $GF(2^n)$ 위에서 균등한 분포에 의해 선택되어진 값일 때, $(a+m_a)m_b$는 a에 독립이다.

Lemma 4) 원소 a가 $GF(2^n)$의 원소이고 원소 λ가 $GF(2^n)$에서의 고정된 상수 값일 때, m_a가 a, λ와 무관한 $GF(2^n)$

위에서 균등한 분포에 의해 선택되어진 값이면 $(a \oplus m_a)^2$, $(a \oplus m_a)^2 \lambda$는 a에 독립이다.

V. 결론

본 논문에서는 AES, ARIA와 같은 블록 암호의 비선형 연산인 역원 연산에 대한 마스킹 기법을 제안하였다. 제안하는 방법은 기존의 복합체 위에서의 역원 연산 방법을 이용하여 설계되었으며 덧셈 마스킹 방법을 적용, 곱셈 마스킹에서의 취약성을 보완하였다. 또한 마스킹 방법의 적용 시 나타날 수 있는 곱셈 연산에서 중복될 수 있는 수식을 찾아 비용적인 측면도 보완하였다.

참고문헌

[1] Advanced Encryption Standard (AES), FIPS PUB 197, November 26, 2001, available at http://csrc.nist.gov/encryption/aes.

[2] B. Zakeri, M. Salmasizadeh, A. Moradi, M. Tabandeh, M. Shalmani, "Compact and Secure Design of Masked AES S-Box", ICICS 2007, LNCS 4861, pp. 216- 229, Springer, 2007.

[3] C. Herbst, E. Oswald, S. Mangard, "An AES Smart Card Implementation Resistant to Power Analysis Attacks," ACNS 2006, LNCS 3989, pp. 239-252, Springer, 2006.

[4] D. Canright, "A Very Compact Rijndael S-box.Technical Report", NPS-MA-04- 001, Naval Postgraduate School (September 2004), http://web.nps.navy.mil/~dcanrig/pub/NPS-MA-05-001.pdf

[5] E. Oswald and K. Schramm. "An Efficient Masking Scheme for AES Software Implementations," WISA 2005, LNCS 3786, pp. 292- 305, Springer, 2006.

[6] E. Oswald, S. Mangard, N. Pramstaller, and V. Rijmen., "A Side-Channel Analysis Resistant Description of the AES S-box," FSE 2005, LNCS 3557, pp. 3- 423, Springer, 2005.

[7] J. Bl " omer, J. Guajardo, and V. Krummel. "Provably Secure Masking of AES," SAC 2004, LNCS 3357, pp. 69- 83, Springer, 2005.

[8] Jovan D. Golic, Christophe Tymen. "Multiplicative Masking and Power Analysis of AES", CHES 2002, LNCS 2523, pp. 198- 212, Springer, 2003.

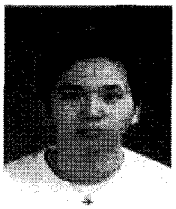
[9] Mehdi-Laurent Akkar and Christophe Giraud. "An Implementation of DES and AES, Secure against Some Attacks", CHES 2001, LNCS 2162, pp. 309-318, Springer, 2001.

[10] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," CRYPTO'99, pp. 388-397, Springer-Verlag, 1999.

[11] P. Kocher, J. Jaffe, and B. Jun, "Introduction to differential power analysis and related attacks," http://www. cryptography.com/dpa/technical, 1998.

[12] P. Kocher, J. Jaffe, and B. Jun, "Timing Attacks on Implementations of Diffie- Hellman, RSA, DSS, and Others Systems." CRYPTO'96, LNCS 1109, pp. 104-113, Springer- Verlag, 1996.

[13] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Power analysis attacks on modular exponentiation in Smart cards," Proc. of Workshop on Cryptographic Hardware and Embedded Systems, pp. 144-157, Springer- Verlag, 1999.



김 희 석 (HeeSeok Kim)

2006년 연세대학교 수학과 졸업. 2008년 고려대학교 정보경영공학전문대학원 공학석사. 2008년 ~현재 고려대학교 정보경영공학전문대학원 박사과정. 관심분야는 부채널 공격, 공개키 암호시스템 안전성 분석 및 고속구현, 타원곡선등 임.

한 동 국 (Dong-Guk Han)

1999년 고려대학교 수학과 졸업. 2002년 고려대학교 수학과 석사(이학석사). 2005년 고려대학교 정보보호대학원 박사(공학박사). 2004년~2005년 일본 Kyushu Univ., 방문연구원. 2005년~2006년 일본 Future Univ.-Hakodate, Post.Doc. 2006년 6월~현재 한국전자통신연구원 정보보호연구단 선임연구원. 관심분야는 암호시스템 안전성 분석 및 고속 구현, 부채널 분석, *RFID/USN* 정보보호 기술등임

조 영 인 (Young In Cho)

2006년 한양대학교 수학과 졸업. 2007년 ~현재 고려대학교 정보경영공학전문대학원 석사과정. 관심분야는 공개키 암호, 암호칩 설계 기술등

홍 석 희 (SeokHie Hong)

1995년 고려대학교 수학과 학사. 1997년 고려대학교 수학과 석사. 2001년 고려대학교 수학과 박사. 1999년~2004년 (주)시큐리티 테크놀로지스 선임연구원. 2003년~2004년 고려대학교 시간강사. 2004년~2005년 K.U. Leuven 박사후연구원. 2005년~현재 고려대학교 정보경영전문대학원 조교수. 관심분야는 대칭키 암호 알고리즘, 공개키 암호 알고리즘, 포렌식등