완전한 형태의 랜덤한 덧셈-뺄셈 체인의 암호분석

한동국*, 장남수*, 정석원*, 박영호**, 김창한***

*고려대학교, 정보보호학과, **세종사이버대학교, 정보보호시스템공학과,
***세명대학교, 정보보호학과

Cryptoanalysis of the Full version Randomized Addition-Subtraction Chains

Dong-Guk Han*, Nam Su Chang*, Seok Won Jung*,
Young-Ho Park**, Chang Han Kim***

*Center for Information and Security Technologies(CIST), KOREA Univ., **Dept of Information Security, Sejong Univ., ***Dept. of Information Security, Semyung Univ.

요 약

Okeya-Sakurai는 [12]에서 단순한 형태의 랜덤한 덧셈-뺄셈 체인의 대응방법[14]은 SPA공격에 취약함을 보였다. 그러나 그들의 분석 방법은 복잡한 형태[14]에는 적용되지 않는다. 본 논문에서는 Okeya-Sakurai의 공격 알고리듬에 두 가지 잠재된 문제가 있음을 보인다. 또한 [12,15]와는 다른 강하고 견고한 새로운 공격 알고리듬을 제안한다. 본 논문에서 제안하는 공격 알고리듬을 사용하면 복잡한 형태의 랜덤한 덧셈-뺄셈 체인 [14] 또한 완벽하게 분석된다. 본 논문의 결과를 표준에서 제안된 163비트로 실험한 결과 단순한 형태에서는 20개의 AD수열로 대략 94%의 확률로 공격이 성공하며 30개의 AD수열로는 대략 99%의 확률로 공격이 성공한다. 또한, 복잡한 형태에서는 40개의 AD 수열로 94%의 확률로 70개의 AD수열로는 99%로의 확률로 공격이 성공한다.

I. 서론

타원곡선 이론을 이용한 공개키 암호시스템(ECC)은 Koblitz[5]와 Miller[9]에 의해 처음으로 제안되었다. 타원곡선은 이산대수문제를 기반으로 하는 공개키 암호시스템이며 다른 암호시스템과 비교하여 작은 키 사이즈로 같은 안전성을 보장한다. 작은 키 사이즈를 가지는 ECC는 저전력으로 구현이 되는 스마트 카드와 같은 장비에 잘 적용된다. 또한 최근에 [7]에서 암호시스템이 구현 된 스마트 카드의 부채널 공격 방법이 소개 되었다. 암호시스템의 부채널 공격 방법으로는 적절한 계산 타이밍 또는 전력 소비량 분석이 있다. 이 두 가지 다른 전력 분석법을 SPA와 DPA라 하며 보호되지 않은 타원곡선 암호시스템에서 스칼라 곱셈 시에 효과가 있다. 전력분석 공격이 타원곡선 암호시스템에 적용되는 기본적인 개념은 Coron에 의하여 [1]에서 소개 되었다. [1]에서 Coron에 의해 소개된 세 가지 전력 분석 공격의 대응방법; 비밀키의 랜덤화, 점의 Blinding, 그리고 사영좌표계의 램덤화이다. 그러나 [11]에서 Okeya-Sakurai는 Coron의 첫 번째와 세 번째 DPA공격 대응 방법은 약간의 문제가 있음을 논했다. 또한, Goubin은 [2]에서 소개된 Coron의 세 번째 전력 분석공격 대응방법을 적용하여도 전력분석 공격이 가능함을 보였다. SPA에 안전한 두 가지 접근방법으로는 덧셈과 두 배 연산을 구별 불가능하게 하는 방법과 덧셈과 두 배 연산을 항상 수행하는 방법이다.

[14]에서 Oswald는 Morain과 Olivos가 제안한 랜덤한 덧셈-뺄셈 체인을 이용하여 이진 스칼라 알고리듬 자체를 랜덤화하는 두 개의 랜덤화된 오토마타를 DPA 대응방법으로 제시하였다. 그러나 Okeya-Sakurai는 [12]에서 랜덤한 덧셈-뺄셈 체인이 SPA 공격에 취약함을 보였다. 그러나 그들은 방법은 단순한 오토마타에만 적용이 가능하다.

본 논문에서는 Okeya-Sakurai 공격방법은 공격자가 비밀값 d의 길이를 알아야한다는 것과 무한원점 연산에서 문제점을 가지고 있음을 제시한다. 또한 미해결 문제로 남아있던 복잡한 버전의 공격이 가능한 강하고 견고한 새로운 공격방법을 제안한다.

II. [12]에서 제안된 알고리즘에서 고려할 사항

Okeya-Sakurai는 [14]의 단순 버전 알고리즘이 덧셈과 뺄셈의 구별 불가능하다는 조건[12]하에도 SPA공격에 취약함을 보였다. 그들의 공격 알고리듬은 **state=0**에서 $d_i=0$이라는 사실을 사용하였다. 그러나 복잡한 버전에서 더 이상 **state=0**에서 $d_i=0$이 아니다. 따라서 복잡한 버전에서 SPA 공격은 해결되지 않은 문제로 남겨져있다. Okeya-Sakurai의 공격 알고리듬은 세 가지 과정으로 간주된다; 스칼라 d의 비트값 결정, AD수열의 자리의 결정, **state**정보의 결정이다.

그러나 그들의 공격 알고리듬[12]은 다음과 같은 두 가지 잠재된 문제가 있다.

▫ 공격자는 비밀값 d의 비트의 길이를 알아야한다 : 공격 알고리즘에서 |d|의 비트 길이가 [12]의 (a)단계에서 항상 제외되어 사용되었다. 본 문제는 공격자에게 두 배 연산으로부터 덧셈을 쉽게 구별할 수 있도록 한다. 스칼라 d의 모든 비트는 두 배 연산을 수행한다. 이것은 두 배 연산의 연산수가 고정되어 있음을 의미한다. 따라서 두 배 연산의 수는 스칼라 d의 비트길이이다.

▫ 그들은 점들의 덧셈에서 점들 중의 하나가 무한원점이라는 사실을 간과했다. : 스칼라 곱셈 dP의 구현에서 $Q=2*O$, $Q=P+O$의 특별한 점들의 연산을 피할 수 있다. 일반적으로 덧셈 혹은 두 배 연산 대신에 점의 복사 등의 방법이 사용되었다. 점의 덧셈에서 점들 중의 하나가 무한원점이라는 사실을 간과한데 있다.

III. 랜덤한 덧셈-뺄셈 체인 대응방법의 분석

단순전력공격에서 공격자는 하나의 스칼라 곱셈의 전력 소비량을 관찰할 수 있음을 가정한다. 공격자의 목표는 완전한 스칼라 곱셈의 전력 트레이스의 관찰로부터 얻은 정보를 이용하여 키를 찾는 것이다. 스칼라 곱셈은 점의 덧셈, 뺄셈 그리고 두 배 연산의 수열로 간주된다. 점의 덧셈 연산은 두 배 연산과는 다른 전력 소비 패턴을 가진다. 점의 덧셈과 뺄셈은 거의 같은 연산이므로 공격자가 구별하는 것이 불가능하도록 구성될 수 있다. 본 논문에서는 점의 덧셈과 두 배가 어느 시점의 전력 소비량 분석에 의하여 구별될 수 있음에 반하여, 점의 덧셈과 뺄셈은 구별 불가능하다. 앞으로는 덧셈과 뺄셈 연산을 덧셈이라 간단히 명명한다. Osward -Aigner[14]에 의해 제안된 랜덤한 덧셈-뺄셈 체인(Addition-Subtraction chain)에서 스칼라 d의 각자리의 0과 1 둘다 덧셈(두 배) 연산에 대응되므로 연산 시 발생하는 전력 소비량의 측정과 비트의 결정과의 관계가 좀더 난해해졌다. 다시 말하면 ,d의 자리가 1(또는 0)일때 어떤 경우에는 덧셈 연산을 하고 또 어떠한 경우에는 두 배 연산을 수행할 것이다. 따라서 [12]에서 Okeya-Sakurai는 랜덤한 오토마타 1을 분석하였으나 랜덤한 오토마타1의 복잡한 형태인 랜덤한 오토마타 2에는 공격이 적용되지 않는다.

그러나 본 논문에서는 전력 소비량 측정에 의한 비트와 연산의 새로운 필요충분조건 관계를 제안한다. 스칼라 곱셈의 전력 소비량의 측정값 n을 사용하여 스칼라 d를 이 관계를 사용하여 어떻게 찾아내는지 보일 것이다. 우리가 제안하는 공격 알고리듬은 이전에 묘사된 취약한 점을 제거하였고, 랜덤한 오토마타 2에 직접 적용된다.

▫ Notation : S를 AD수열이라 하면; 수열은 덧셈(Addition) A와 두배(Doubling) D로 간주된다. $S_i[k]$는 i번째 수열의 k번째 값을 뜻하며 $S_i[k_i, n]$은 i번째 수열의 k_i번째부터 연속된 n개의 값을 뜻한다. |S|는 A, D 문자의 수이며 비어있는 $S_i[k]$는 $S_i[k]$=NULL이라 한다.

1. 랜덤한 덧셈-뺄셈 체인 대응방법의 특징

본 소절에서는 Oswald-Aigner가 제안[14]한 랜덤한 덧셈-뺄셈 체인의 특성을 살펴보자. 우리는 제안된 표 1, 2, 3으로부터 키의 비트와 AD수열 값의 관계를 유도한다.

▫ 가정 : 공격자는 랜덤화된 덧셈-뺄셈 체인 대응방법이 적용된 암호화 장비에 타원곡선의 점을 입력하고 AD수열을 얻는다. 공격자는 n번의 반복 수행에서 n개의 AD수열을 얻는다. 본 논문에서는 Okeya-Sakurai의 [12]에서의 표기법을 따른다.

■ **Property 1.** 만약 모든 i_1에 대해 $S_{i_1}[k_{i_1}]$ =A, $S_{i_1}[l]$=D $(0\le l\le k_{i_1}-1)$이고 어떤 i_2, i_3에 대

하여 $S_{i_2}[k_{i_2}+2]$=A, $S_{i_3}[k_{i_3}+2]$=D이면 d_{j-1}=1, d_j=1, 그리고 $\{d_0, \cdots, d_{j-2}\}$는 j>1일때 모두 0이다.

■ **Property 2.** 만약 모든 i_1에 대해 $S_{i_1}[k_{i_1}]$=A, $S_{i_1}[l]$=D $(0 \le l \le k_{i_1}-1)$이고 모든 $i_2(i_3)$에 대하여 $S_{i_2}[k_{i_2}+2]$=A(또는 $S_{i_3}[k_{i_3}+2]$=D)이면 d_{j-1}=0, d_j=1이다. 만약 j=2이면 d_0=1이고 그렇지 않으면 단하나만 d_l=1 $(0 \le l \le j-2)$ 이고 나머지는 모두 0이다.

이 두 가지 성질은 점의 덧셈에서 하나의 점이 무한원점인 경우에서 유도된다. P가 무한원점이고 다음 비트가 1인 경우 덧셈 연산은 수행되지 않고 단지 두 배 연산만 수행된다.

■ **Theorem 1.** 만약 현재 **state=0** 또는 **state =110**일 때 키의 비트 d_i=0일 필요충분조건은 모든 i_1에 대해 $S_{i_1}[k_{i_1}]$=D일 때 이다.

■ **Theorem 2.** 만약 현재 **state=0** 또는 **state =110**일 때 키의 비트 d_i=1일 필요충분조건은 모든 i_1에 대해 P가 무한원점이 아닐 때 $S_{i_1}[k_{i_1}]$=A 이다.

d_{j-2}, d_{j-1}	d_j, d_{j+1}	AD sequence	Probability
11 or 01	11	DAAD	1/4
		DDA	1/8
		DD	1/8
		ADDA	1/8
		ADD	1/8
		ADAD	1/4
	10	DAD	1/2
		ADAD	1/4
		ADD	1/4
	01	ADAD	1/2
		DAD	1/2
	00	ADD	1/2
		DD	1/2

표 1. 연속된 비트가 11 또는 01일 때의 AD수열

표 1.은 이전의 연속된 비트가 11 또는 01이며 무한원점의 아닐 때 d_j, d_{j+1}의 계산 후의 AD수열과 확률값을 나타낸 것이다.

■ **Property 3.** 만약 어떤 i_1, i_2에 대해 $S_{i_1}[k_{i_1}]$=D와 $S_{i_2}[k_{i_2}]$=A이면 이전의 두 비트는 d_{j-2}=1, d_{j-1}=1이거나 또는 j-2번째 **state=110**일 때 d_{j-2}=0, d_{j-1}=1이다.

■ **Theorem 3.** 만약 어떤 i_1, i_2에 대해 $S_{i_1}[k_{i_1}]$=D와 $S_{i_2}[k_{i_2}]$=A일 필요충분조건은 이전의 두 비트는 d_{j-2}=1, d_{j-1}=1이거나 또는 j-2번째 **state=110**일 때 이전의 두 비트는 d_{j-2}=0, d_{j-1}=1, 일 때이다. 필요한 상태의 오류 확률은 (2/3)"보다 작을 때이다.

d_{j-2}, d_{j-1}	d_j, d_{j+1}, d_{j+2}	AD sequence	Probability
11 or 01	111	ADDAAD	1/8
		ADDDA	1/16
		ADDD	1/16
		ADADDA	1/16
		ADADD	1/16
		ADADAD	1/8
		DAADDA	1/16
		DAADD	1/16
		DAADAD	1/8
		DDAAD	1/8
		DDDA	1/16
		DDD	1/16
	110	ADDAD	1/4
		ADADAD	1/8
		ADADD	1/8
		DAADAD	1/8
		DAADD	1/8
		DDAD	1/4

표 2. 연속된 비트가 11 또는 01이며 d_j=1일때 AD수열

d_{j-3}, d_{j-2}, d_{j-}	d_j, d_{j+1}	AD sequence	Probability
110	11	ADAD	2/3
		ADDA	1/6
		ADD	1/6
	10	ADD	2/3
		ADAD	1/3
	01	DAD	1
	00	DD	1

표 3. 연속된 비트가 110일때 AD수열

2. 제안하는 공격 알고리듬

공격자는 랜덤한 덧셈-뺄셈 체인 대응방법이 적용된 암호화 장비에 타원곡선의 점을 입력하고 AD수열을 얻는다. 공격자는 n번의 반복 수행에서 n개의 AD수열을 얻고, 아래와 같이 비밀값 d를 획득한다.

본 논문에서 제안하는 알고리즘은 두 과정으로 분류된다. 한 과정은 비밀값 d의 비트들을 결정하는 것이고, 또 다른 과정은 S의 값의 위치를 결정하는 것이다. 본 공격 알고리듬은 Okeya-Sakurai의 것과 달리 state의 정보를 필요로 하지 않는다.

■ 랜덤화된 오토마타 2의 공격 알고리듬 : 랜덤한 오토마타2의 공격 알고리듬은 다음 4가지 알고리즘으로 구성된다.

-초기화 : $k_i=0$이고 $j=0$이며; 키 검출 알고리듬을 수행한다. k_i는 AD수열의 위치와 관련이 있고 j는 스칼라 d의 비트와 관련이 있다.

키 검출 알고리듬 0

▸**만약 어떤 i_1에 대하여** $S_{i_1}[k_{i_1}]$=NULL**이면** d_j=1, 나머지 비트는 0으로 설정되며 알고리듬 종료

▸**만약 모든 i_1에 대하여** $S_{i_1}[k_{i_1}]=A$**이고 어떤 i_2에 대하여** $S_{i_2}[k_{i_2}+2]$=NULL**이면** d_j=1이고 d_l은 $0\le l\le j-1$에서 하나만 1이고 나머지는 0이다. j번의 조사를 통해 d_l을 결정, 알고리즘 종료

▸**만약 모든 i_1에 대하여** $S_{i_1}[k_{i_1}]=D$**이면** $j\leftarrow j+1$, 모든 i에 대해 $k_i\leftarrow k_i+1$, 알고리듬0 수행

▸**만약 모든 i_1에 대하여** $S_{i_1}[k_{i_1}]=A$**이고**

• 만약 어떤 i_2와 i_3에 대하여 $S_{i_2}[k_{i_2}+2]$=A이고 $S_{i_3}[k_{i_3}+2]$=D이면 $d_{j-1}=1$, $d_j=1$, $j>1$에 대해 $\{d_0,\cdots,d_{j-2}\}$는 모두 0, $j\leftarrow j+1$ 모든 I에 대해 $k_i\leftarrow k_i+2$, 알고리듬1-1 수행

• 아니면 $d_{j-1}=0$, $d_j=1$. 만약 j=2이면 $d_0=1$, 아니면 d_l은 $0\le l\le j-2$에서 하나만 1이고 나머지는 0이다. j-1번의 조사를 통해 d_l을 결정, $j\leftarrow j+1$, 모든 i에 대해 $k_i\leftarrow k_i+2$, 알고리듬1 수행

키 검출 알고리듬 1

▸**만약 어떤 i_1에 대하여** $S_{i_1}[k_{i_1}]$=NULL**이면** $d=\sum_{k=0}^{j-1} d_k 2^k$이며 알고리듬 종료

▸**만약 모든 i_1에 대하여** $S_{i_1}[k_{i_1}]=D$**이면** d_j=0, $j\leftarrow j+1$; 모든 i에 대해 $k_i\leftarrow k_i+1$, 알고리듬1 수행

▸**만약 모든 i_1에 대하여** $S_{i_1}[k_{i_1}]=D$**이면** $j\leftarrow j+1$, 모든 i에 대해 $k_i\leftarrow k_i+1$, 알고리듬1 수행

▸**아니면**

• 만약 모든 i_1에 대하여 $S_{i_1}[k_{i_1}]=A$이면 d_j=1 $j\leftarrow j+1$, 모든 i에 대해 $k_i\leftarrow k_i+2$, 알고리듬1 수행

• 아니면 알고리듬 1-1 수행

키 검출 부분 알고리듬 1-1

▸**만약 어떤 i_1에 대하여** $S_{i_1}[k_{i_1}]$=NULL**이면**

$d=\sum_{k=0}^{j-1} d_k 2^k$이며 알고리듬 종료

▸**만약 어떤 i_2에 대하여** $S_{i_2}[k_{i_2}+2]$=NULL**이면**

d_j=1이고 $d=\sum_{k=0}^{j} d_k 2^k$이며 알고리듬 종료

▸**만약 모든 i_1, i_2에 대해** $S_{i_1}[k_{i_1},3]$=DAA**이고** $S_{i_2}[k_{i_2}+5]$=NULL**이면** $d_j=1$, $d_{j+1}=1$이고 $d=\sum_{k=0}^{j} d_k 2^k$. 알고리듬 종료.

▸**만약 어떤 i_1에 대해** $S_{i_1}[k_{i_1},3]$=DAA**이면**

• 만약 어떤 i_2에 대해 $S_{i_2}[k_{i_2},4]$=DDAD 또는 $S_{i_2}[k_{i_2},5]$=ADDA

D이면; $d_j=1$, $d_{j+1}=1$, d_{j+2}=0이고 $j\leftarrow j+1$, 모든 i에 대해 $S_i[k_i,4]$=DDAD이면 $k_i\leftarrow k_i+4$, 모든 i에 대해 $S_i[k_i,5]$=DAADD, ADDAD, 또는 ADADD이면 $k_i\leftarrow k_i+5$, 다른 경우이면 $k_i\leftarrow k_i+6$, 부분 알고리듬1-1 수행

• 아니면 $d_j=1$, $d_{j+1}=1$ (d_{j+2}=1), $j\leftarrow j+2$; 만약 모든 i에 대해 $S_{i_1}[k_{i_1},3]$=DDD이면 $k_i\leftarrow k_i+2$, 만약 모든 i에 대해 $S_{i_1}[k_{i_1},3]$=DDA 또는 $S_{i_1}[k_{i_1},4]$=ADDD이면 $k_i\leftarrow k_i+3$ 다른 경우이면 $k_i\leftarrow k_i+4$ 부분 알고리듬 1-1 수행

▸**아니면**

• 만약 어떤 i_2에 대해 $S_{i_2}[k_{i_2},3]$=DAD이면

-만약 어떤 i_2에 대해 $S_{i_2}[k_{i_2},3]$=ADD이면 $d_j=1$, $d_{j+1}=0$, $j\leftarrow j+2$; 모든 i에 대해 $S_i[k_i,4]$=ADAD이면 $k_i\leftarrow k_i+4$, 다른 경우이면 $k_i\leftarrow k_i+3$, 부분 알고리듬 1-2 수행

-아니면 $d_j=0$, $d_{j+1}=1$, $j\leftarrow j+2$; 모든 i에대해 $S_i[k_i,3]$=DAD이면 $k_i\leftarrow k_i+3$, 다른 경우이면 $k_i\leftarrow k_i+4$; 부분 알고리듬 1-1 수행

• 아니면 $d_j=0$, $d_{j+1}=0$, $j\leftarrow j+2$; 모든 i에대해 $S_i[k_i,2]$=DD이면 $k_i\leftarrow k_i+2$, 다른 경우이면 $k_i\leftarrow k_i+3$; 알고리듬 1 수행

키 검출 부분 알고리듬 1-2

▸**만약 어떤 i_1에 대하여** $S_{i_1}[k_{i_1}]$=A**이면**; d_j= 1, $j\leftarrow j+1$; 모든 i에 대해 $k_i\leftarrow k_i+2$, 부분 알고리듬 1-1 수행.

▸**만약 모든 i_1에 대하여** $S_{i_1}[k_{i_1}]=D$**이면** d_j=0, $j\leftarrow j+1$; 모든 i에 대해 $k_i\leftarrow k_i+1$, 알고리듬1 수행

▸**만약 모든 i_1에 대하여** $S_{i_1}[k_{i_1}]=D$**이면** $j\leftarrow j+1$, 모든 i에 대해 $k_i\leftarrow k_i+1$, 알고리듬1 수행

▸**아니면**; d_j=0, $j\leftarrow j+1$; 모든 i에 대해 $k_i\leftarrow k_i+1$, 알고리듬 1 수행.

2. 구현 결과

본 논문의 공격 알고리듬을 랜덤화된 오토마타 1과 2에 적용하였다. 표준에서 제안된 163비트에 적용한 구현 결과는 그림1과 같다. 그림1로부터

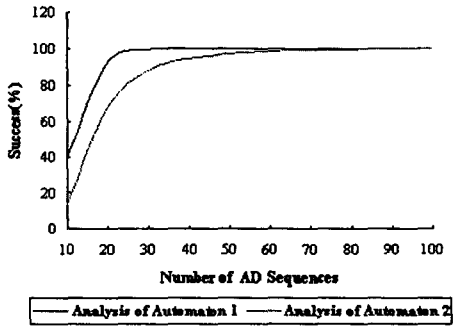

그림 1 랜덤한 오토마타 1과 2의 비교

단순한 오토마타 1은 AD수열 20개와 30개로 각각 대략 94%와 99%의 확률로 공격이 성공한다. 그리고 복잡한 오토마타 2는 AD수열 40개와 70개로 각각은 대략 94%와 99%의 확률로 공격이 성공한다. 예를 들어 AD수열 30개로의 99%의 성공 확률은 공격자가 같은 비밀값에 대하여 AD수열 30개로 100번을 반복하였을 때 대략 99번 비밀값을 검출할 수 있음을 뜻한다. 우리는 실제로 많은 다른 비밀값에 대해서 테스트해 보았다.

참고문헌

[1] J.S.Coron, Resistance against Differential Power Analysis for Elliptic Curve Crytosystems, *In Workshop on Cryptographic Hardware and Embedded Systems*(CHES'99), LNCS 1717,(1999),292-302.

[2] L.Goubin, A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems, *Public Key Cryptography* (PKC 2003), LNCS2567, (2003), 199-211.

[3] M.Joye and J.Quisquater, Hessian ellitpic curves and side-channel attacks, *In Workshop on Cryptographic Hardware and Embedded Systems*(CHES'01), LNCS2162, (2001), 402-410.

[4] M.Joye and C.Tymen, Protections against differential analysis for elliptic curve cryptography, *In Workshop on Cryptographic Hardware and Embedded Systems*(CHES'01), LNCS2162, (2001),377-390.

[5] N.Koblitz, Elliptic curve cryptosystems, *In Mathematics of Computation*, volume 48, (198 7), 203-209.

[6] P.Kocher, Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems, *In Advances in Cryptology*- CRYPTO'96, LNCS1109,(1996),104-113.

[7] P.Kocher, J.Jaffe, and B.Jun, Differential power analysis, *In Advances in Cryptology*- CRYPTO'99, LNCS1666,(1999),388-397.

[8] P.Liardet and N.Smart, Preventing SPA/DPA in ECC systems using the Jacobi form, *In Workshop on Cryptographic Hardware and Embedded Systems*(CHES'01), LNCS2162, (2001), 391-401.

[9] V.S.Miller, Use of elliptic curves in cryptography, *In Advances in Cryptology*- CRYPTO '85, LNCS218,(1986), 417-426.

[10] F.Morain and J.Olivos, Speeding up the computation on an elliptic curve using addition-subtraction chains, *Inform Theory Appl.*, vol 24, (1990), 531-543.

[11] K.Okeya, K.Sakurai, Power analysis breaks elliptic curve cryptosystems even secure against the timing attack, *Indocrypt* 2000, LNCS 1977, (2000),178-190.

[12]K.Okeya, K.Sakurai, On Insecurity of the Side Channel Attack Countermeasure Using Addition-Subtraction Chains under Distinguishability between Addition and Doubling, *Information Security and Privacy* (ACISP'02), LNCS2384, (2002),420-435.

[13]K.Okeya, H.Kurumatani and K.Sakurai, Elliptic curves with the Montgomery form and their cryptographic applications, *Public Key Cryptography* (PKC 2000), LNCS1751, (2000), 446-465.

[14]E.Oswald, M.Aigner, Randomized Addition-Subtraction Chains as a Countermeasure against Power Attacks}, In Workshop on Cryptographic Hardware and Embedded Systems (CHES'01), LNCS2162, (2001),39-50.

[15]C.D. Walter, Security Constraints on the Oswald-Aigner Exponentiation Algorithm, *Cryptology ePrint Archive*, Report 2003/013, (2003).http//eprint.iacr.org/.