보안공학연구논문지 *Journal of Security Engineering*
Vol.14, No.1 (2017), pp.43-58
http://dx.doi.org/10.14257/jse.2017.02.05

암호용 디바이스에 대한 구현 공격에 강인한 이중 멱승 알고리듬

박은수[1], 하재철[2]

Double Exponentiation Algorithm Resistant to the Implementation Attacks on Cryptographic Device

Eunsoo Park[1], Jaecheol Ha[2]

요 약

정보보호용 디바이스에 RSA 시스템과 같은 암호 알고리듬을 그대로 탑재하여 사용할 경우 전력 분석이나 오류 주입 등과 같은 구현 공격에 의해 비밀 키가 노출될 수 있다. 본 논문에서는 덧셈 체인에 기반한 이중 멱승 알고리듬이 입력 메시지에 대한 오류 주입 공격에 취약하며 덧셈 체인을 구해야 하는 비효율성을 지적하고 이를 해결할 수 있는 새로운 이중 멱승 알고리듬을 제안한다. 제안하는 이중 멱승 알고리듬은 기존에 제시된 전력 분석 공격 및 오류 주입 공격을 방어할 수 있으며 안전한 RSA-CRT(RSA based on Chinese Remainder Theorem) 시스템을 구현하기 위해 효과적으로 활용할 수 있다.

핵심어 : 암호용 단말 장치, 구현 공격, 암호 알고리듬, 이중 멱승 연산, 안전성

Abstract

When some cryptographic algorithms used in cryptosystem such as RSA are naively implemented on a security device, the secret key of cryptosystem can be exposed to an attacker by the power analysis and fault injection attacks In this paper, we point out that the double addition chain exponentiation algorithm is vulnerable to the fault injection attack on input message and has inefficiency due to the computing of addition chain. We proposed a novel double exponentiation algorithm, which defeats most previous power analysis and fault injection attacks and can be adopted for secure RSA-CRT implementation.

Keywords : Cryptographic Device, Implementation Attack, Encryption Algorithm, Double Exponentiation, Security Evaluation

Received(January 07, 2017), Review request(January 08, 2017), Review Result(1st: January 23, 2017)

Accepted(February 06, 2017), Published(February 28, 2017)

[1]Dept. of Information Security, Hoseo University, Asan-si, Chungnam-do, 31499, Korea
email: pespskk@gmail.com
[2](Corresponding Author) Dept. of Information Security, Hoseo University, Asan-si, Chungnam-do, 31499, Korea
email: jcha@hoseo.edu

* 이 논문은 2016년도 호서대학교의 재원으로 학술연구비 지원을 받아 수행된 연구임 (2016-0039)

ISSN: 1738-7531 JSE

1. 서론

스마트 카드와 같은 정보보호용 디바이스를 이용하여 데이터를 암호화 하거나 디지털 서명을 수행할 경우, 필요한 암호 알고리듬들을 전용 프로세서에 직접 구현하여 사용하기도 한다. 그리고 사용자는 정보보호에 필요한 큰 길이의 비밀 키를 디바이스 내부에 저장한 후 이 키를 이용하여 암호화 및 서명 연산을 수행하게 된다. 이때 사용하는 암호 알고리듬들은 이론적으로 안전성이 증명된 것을 사용하지만 이 알고리듬을 하드웨어 장치에 구현하는 과정에서 발생하는 오류나 문제점으로 인해 내부의 비밀 키가 노출될 수 있음이 밝혀졌다[1]. 이와 같은 물리적인 결함을 이용하는 공격을 암호 시스템에 대한 구현 공격(implementation attack)이라고 한다. 구현 공격은 크게 정보보호 디바이스가 동작하는 과정에서 발생하는 전력이나 전자기파 등을 측정한 후 이를 분석함으로써 비밀 키를 찾아내는 수동적인 공격 방법과 공격자가 의도적으로 디바이스에 오류를 주입하여 비밀 키를 찾아내는 능동적인 공격 방법으로 나눌 수 있다. 부채널 공격(side channel attack)이라고도 불리는 수동적 공격 방법 중 소비되는 전력을 분석하는 공격 형태를 전력 분석(power analysis) 공격이라 하는데 이를 크게 단순 전력 분석(Simple Power Analysis, SPA) 공격과 차분 전력 분석(Differential Power Analysis, DPA) 공격으로 나누기도 한다[2]. 능동적인 부채널 공격 방법 중 오류 주입 공격(Fault Attack, FA)은 Boneh 등에 의해 처음으로 제안된 공격으로서 암호 알고리듬을 수행하는 동안 암호 연산용 디바이스에 고의적으로 오류를 주입한 후, 그 출력을 분석함으로써 내장된 비밀 키를 찾아내는 공격 기법이다[3].

이러한 구현 공격들은 AES(Advanced Encryption Standard)[4]와 같은 표준 블록 암호 시스템은 물론 RSA(Rivest, Shamir, and Adelman)와 같은 범용 공개 키 암호 시스템[5]에도 쉽게 적용 가능하다. 특히, RSA 암호 시스템에서는 비밀 키를 이용한 멱승(exponentiation)[6] 연산이 매우 중요한데 이 멱승 알고리듬의 수행 과정이 공격의 주된 목표가 되고 있다. 따라서 구현 공격에 대응할 수 있고 암호용 멱승 알고리듬 설계가 디바이스의 안전성을 보증하기 위한 중요한 이슈가 되고 있다.

현재까지 멱승 알고리듬에 대한 수동적 구현 공격으로는 SPA, DPA[7], Doubling 공격[8], 충돌 전력 분석(Correlation Power Analysis, CPA) 공격[9] 등이 있었다. 최근에는 모듈라 곱셈 연산의 입·출력에 대한 전력 파형의 충돌 쌍을 분석하여 비밀 키를 찾아내는 수평 충돌 전력 분석(Horizontal CPA) 공격[10][11]이 제안되기도 하였다. 오류 주입 공격으로는 연산 과정에서 비트 단위로 오류를 주입하는 비트 오류 주입 공격[12]과 C-safe 오류 공격[13] 등이 대표적이다. 특히, RSA 암호 시스템 연산 시 계산 효율을 높이기 위해 사용하는 RSA-CRT(RSA based on Chinese Remainder Theorem) 알고리듬[14]은 Bellcore 공격이라고 불리는 오류 주입 공격에 매우 취약한 특성을 보이고 있다[3]. 또한, 최근에는 오류 주입 공격과 SPA를 결합한 조합 공격(Passive and Active Combined Attack, PACA)이 제안되기도 하였다[15][16].

구현 시스템에 대한 다양한 물리적 공격들이 등장함에 따라 이를 막기 위한 대응책도 연구되어 왔는데, 특히 SPA와 C-safe 오류 공격을 동시에 방어할 수 있는 효율적인 이진(binary) 멱승 알고리듬에 관한 연구가 많았다. 본 논문에서는 안전한 RSA 멱승 알고리듬을 구현하기 위한 덧셈 체인(addition chain)에 기반한 이중 멱승(double exponentiation) 알고리듬[17]을 살펴보고 공격 취약성 및 구현 효율성을 분석해 본다. 또한, 지금까지 제시되었던 구현 공격들을 모두 방어할 수 있는 이중 멱승에 기반한 새로운 멱승 알고리듬을 제안하고자 한다. 특히, 제안하는 이중 멱승 알고리듬은 블라인딩(blinding) 기법을 적용하여 안전한 RSA-CRT를 구현하는데 활용할 수 있다.

2. RSA 암호 시스템에 대한 구현 공격 및 대응 기법

2.1 RSA 멱승 알고리듬

데이터 암호화나 디지털 서명을 위해 많이 사용하고 있는 RSA 암호 시스템에서는 다음과 같이 메시지 M에 대해 지수(exponent) d의 멱승 $M^d \bmod N$ 연산이 필요하다. 여기서 N은 두 소수 p와 q의 합성수를 나타낸다. 특히, 암호문에 대한 복호나 메시지에 대한 디지털 서명을 수행할 때에는 암호용 디바이스에 저장되었던 비밀 키 d를 지수로 사용한다. 여기서 비밀 키 d는 l비트의 길이를 가지는 정수라고 가정한다.

멱승 알고리듬은 지수로 사용되는 비밀 키 d를 어떤 단위로 탐색하여 처리하는가에 따라 이진 방식(binary method), m진 방식(m-ary method) 그리고 윈도우 방식(window method) 등이 있지만 구현의 효율성과 개발 환경을 고려하여 이진 방식이 많이 사용되고 있다[6]. 또한, 멱승 방법들은 비밀 키를 처리하는 방향에 따라 최하위 자리부터 처리하는 Right-to-Left 방식과 최상위 자리부터 처리하는 Left-to-Right 방식으로 나누어진다. 고전적인 이진 멱승 알고리듬에서는 d를 한 비트씩 탐색하면서 d_i비트가 1일 경우에는 곱셈(multiply)과 자승(square)을 수행하며 그렇지 않은 경우에는 자승만 수행하게 된다.

고전적인 이진 멱승 알고리듬은 SPA 공격에 의해 쉽게 비밀 키가 노출될 수 있어 SPA에 대응하는 Square-and-Multiply-Always 알고리듬이 제안되기도 하였다[7]. 특정 메시지 M에 대한 멱승 $M^d \bmod N$을 수행하는 Right-to-Left 형태의 Square-and-Multiply-Always 알고리듬을 나타낸 것이 [그림 1]이다. 그림의 단계 2의 반복문을 보면 비밀 키 비트 값과 관계없이 모듈라 자승 한 번과 모듈라 곱셈 한 번을 정규적으로 수행하고 있다.

RSA 암호 시스템을 구현할 경우 계산 효율성을 높이기 위해 중국인의 나머지 정리에 기반한 RSA-CRT 기법을 사용한다. RSA-CRT 암호 기법은 일반 RSA 방식보다 이론적으로 약 1/4까지 연산량을 줄일 수 있는 것으로 알려져 있다. RSA-CRT 계산 과정에서는 소수 p와 q를 모듈러스로 하는 두 번의 멱승, 즉 $S_p = M^{d_p} \bmod p$와 $S_q = M^{d_q} \bmod q$ 연산을 수행한 후 이 결과를 아래와 같이 재결합(recombination)하여 출력(서명문 혹은 복호문) S를 만들게 된다.

$$S = CRT(S_p, S_q) = ((S_q - S_p) \cdot I_p \mod q) \cdot p + S_p \quad (1)$$

여기서 d_p와 d_q는 $d_p = d \bmod (p-1)$과 $d_q = d \bmod (q-1)$를 나타내며, I_p는 모듈러스 q에 대한 p의 역원인 $I_p = p^{-1} \bmod q$을 의미한다.

Square-and-Multiply-Always exponentiation : SMA-Exp(M, N, d) 입력 : M, N, d 출력 : $M^d \bmod N$
1. $S = 1$, $R = M$ 2. for $i = l-1$ down to 0 { 2.1 $S = S \cdot S \bmod N$ 2.2 if($d_i \equiv 1$) $S = S \cdot M \bmod N$ 2.3 else $T = S \cdot M \bmod N$ } 3. Return(S)

[그림 1] Square-and-Multiply-Always 멱승 알고리듬

[Fig. 1] Square-and-Multiply-Always exponentiation algorithm

2.2 구현 공격 및 대응책

상기한 이진 멱승 알고리듬이나 Square-and-Multiply-Always 방법은 구현 공격에 대한 대응 방안이 거의 고려되지 않은 것으로 이 알고리듬들을 그대로 정보보호용 디바이스에 구현할 경우에는 물리적 구현 공격에 취약하여 비밀 키를 노출하게 된다. 지금까지 제시된 대표적인 구현 공격과 그에 따른 대응 방안을 정리하면 다음과 같다.

1) SPA 공격

고전적인 이진 멱승 알고리듬을 사용하여 멱승을 구현할 경우, 공격자는 암호 디바이스를 구동할 때 측정한 전력 파형을 관찰하는 것만으로도 비밀 키를 찾아낼 수 있는데 이를 SPA 공격이라 한다. 공격자는 한 개의 전력 파형만 얻을 수 있고 그 파형을 통해 자승 연산과 곱셈 연산을 구별할 수 있으면 순차적으로 모든 비밀 키 비트 d_i를 추출할 수 있다.

SPA 공격에 대응하기 위해 Square-Multiply Always 이진 방식이나 Montgomery Ladder 방식[18]이 제안되기도 하였는데 이 대응책들은 비밀 키 비트와 관련한 반복문 연산을 할 경우 정규적으로 자승과 곱셈을 한 번씩 수행하도록 알고리듬을 설계하였다. SPA 대응 알고리듬을 설계할 때 주의할 점은 최종적인 멱승 결과와 연관성이 없는 더미(dummy) 연산을 정규화를 위해 사용하면 이 더미 연산이 C-safe 오류 공격에 취약하다는 것이다. 따라서 멱승 알고리듬에 따라 더미 연산이 있는 경우에는 오류 주입 여부를 검사할 수 있어야 한다.

2) DPA 공격

DPA 공격은 하나의 파형만을 사용하는 SPA와 달리 수십~수백 개의 전력 파형을 수집한 후 이들을 통계학적 기법을 이용하여 비밀 키를 찾아내는 기법이다. 따라서 정규적 특성을 이용하는 SPA 대응 알고리듬이라 할지라도 DPA 공격에 취약할 수 있다. DPA 공격에 대응하기 위해서는 지수로 사용하는 비밀 키를 랜덤화하거나 메시지 혹은 모듈러스를 랜덤화하는 것과 같은 블라인딩 기법을 사용한다[7]. 그러나 DPA 공격에 대한 대응책 중에서 특정 랜덤화 기법 하나만 사용할 경우에는 또 다른 구현 공격의 대상이 될 수 있으므로 종합적인 DPA 대응책이 사용되어야 한다.

3) (Relative) Doubling 공격

Doubling 공격에서 공격자는 특수한 관계를 가진 두 메시지를 입력하고 얻은 두 개의 멱승 전력 파형을 분석하여 비밀 키를 찾아내는 공격이다[8]. 여기서 입력되는 두 메시지는 M과 M^2 관계를 가지고 있으며 Left-to-Right 형태의 이진 멱승 방식에 적용할 수 있다. 이 외에도 SPA에 대응할 수 있는 Montgomery Ladder 멱승 알고리듬은 Relative Doubling 공격[19]에 의해 비밀 키가 노출될 수 있음이 밝혀졌다. Relative Doubling 공격은 특수한 관계를 가진 두 메시지를 입력으로 얻은 전력 파형을 분석하여 인접한 비밀 키 비트의 연관성을 찾아 비밀 키를 추출하는 공격 기법이다.

4) Horizontal CPA 공격

Witteman 등은 Square-Multiply Always 멱승 알고리듬에 대한 Horizontal CPA 공격을 제안하였는데[10] 이 공격은 멱승 연산 시 수행되는 곱셈과 자승간의 소비 전력량을 측정한 후 상호 상관도(cross-correlation)를 분석하여 비밀 키를 찾아내는 공격 방법이다. 즉, Horizontal CPA 공격에서는 곱셈이나 자승을 수행하는 과정에서 입력으로 동일한 피승수를 사용하는지 여부를 수집된 전력 파형간의 상관도 분석을 통해 알 수 있다면 사용된 비밀 키 비트를 추출할 수 있다. Witteman 등의 공격에서는 두 연산의 상관도 분석을 통해 비밀 키 비트가 0인지 1인지 직관적으로 판별하는 기법을 사용하며, Feix 등이 제시한 공격[11]에서는 인접한 비밀 키 비트가 같은 값인지를 판단하는 기법을 사용한다.

5) C-safe 오류 공격

C-safe 오류 공격은 멱승 알고리듬이 수행되는 중간에 오류를 주입 후 그 결과가 정상인지 그렇지 않은지 여부를 판별하여 비밀 키 비트를 찾아내는 공격이다[13]. 예를 들어 특정한 비밀 키 비트와 관련한 연산이 최종 결과 값에 영향을 미치지 않는 더미 연산이 있으면 이 연산 과정에 오류를 주입해도 출력 결과가 정상이 된다. 따라서 이러한 더미 연산에 오류를 주입 후 그 결과가 정상이면 오류를 주입했던 연산과 관련한 비밀 키 비트를 찾아내는 원리를 이용한다. 그러므로 멱승 알고리듬에서는 비밀 키와 직접 관련 없는 더미 연산을 제거하거나 더미 연산의 오류 주입 여부를

검사하는 기능을 추가함으로써 C-safe 오류 공격을 방어할 수 있다.

5) 오류 주입 및 SPA 조합 공격

오류 주입 공격과 SPA 공격을 결합한 조합 공격은 비밀 키와 관련된 여러 개의 레지스터 중 하나를 0으로 만든 후 연산을 수행하는 과정의 단순 전력 파형을 분석하는 방법이다[15]. 즉, 멱승 연산 시 의도적으로 0으로 세팅된 레지스터와 모듈라 곱셈을 수행하는 연산 과정이 있음을 전력 파형 분석을 통해 구별할 수 있으면 모든 비밀 키를 모두 찾아낼 수 있다. 최근 연구에 의하면 전력 분석 공격이나 오류 주입 공격에 강인한 특성을 가지며 RSA-CRT 구현에 효과적인 것으로 알려진 BNP(Boscher, Naciri, and Prouff) 알고리듬[20]이나 BHT(Boscher, Handschuh, and Trichina) 멱승 알고리듬[21]도 이 조합 공격에 의해 비밀 키가 노출될 수 있음이 밝혀졌다[16]. 조합 공격에 대응하기 위해서는 특정 레지스터가 공격자의 오류에 의해 0으로 초기화되는 것을 방지하거나 지수 블라인딩 기법을 이용해 오류가 주입되어도 SPA 공격이 불가능하도록 설계하여야 한다.

7) Bellcore 오류 주입 공격

1997년 Boneh 등에 의해 RSA-CRT에 대한 오류 주입 공격이 처음 제안되었는데 이를 Bellcore 오류 주입 공격이라 부른다[3]. 이 오류 주입 공격은 RSA-CRT 연산 시 수행되는 두 번의 멱승 결과인 S_p나 S_q 중의 어느 하나에 오류를 주입한 잘못된 서명 S'을 얻을 수 있을 경우 쉽게 N을 소인수 분해하여 비밀 키를 찾아내는 공격이다. 즉, 동일한 하나의 메시지에 대해 정상 서명 값이 S이고, S_p에 오류가 주입되어 생성된 오류 서명 S'을 얻을 수 있으면 $q = GCD(S - S', N)$을 계산함으로써 비밀 소수 q를 구할 수 있다. 이 공격은 알고리듬을 수행하는 과정에서 오류를 주입할 수 있는 시간도 길고 공간적 범위도 넓어 RSA-CRT에 대한 매우 위협적인 공격 방법으로 알려져 있다.

3. 이중 멱승을 이용한 오류 주입 공격 대응 기법

2009년 Rivain은 RSA-CRT에 대한 오류 주입 공격에 대응하기 위해서 이중 덧셈 체인(double addition chains)을 이용한 멱승 기법을 제안하였다[17]. 제안된 기법의 핵심 원리는 두 개의 지수 d와 $\phi(N) - d$에 대한 멱승 쌍 $(M^d \bmod N,\ \ M^{2\phi(N)-d} \bmod N)$을 계산하기 위해 이중 멱승 계산을 수행하고 계산된 두 멱승 값들이 $M^d \cdot M^{\phi(N)-d} \equiv 1 \bmod N$ 를 만족하는지 확인함으로써 오류 주입 여부를 검사하는 방법이다. 여기서 $\phi(N)$은 N의 Euler의 totient 함수이다. 한편, Rivain의 덧셈 체인을 구하는 알고리듬에서는 입력되는 지수 쌍 (a,b)가 $a \le b$를 만족해야 한다. 따라서 오일러의 정리(Euler's theorem)에 의해 $M^{\phi(N)-d} \equiv M^{2\phi(N)-d} \bmod N$을 만족하기 때문에 두 개의 지수 d와 $2\phi(N) - d$에 대한 멱승 쌍 $(M^d \bmod N,\ \ M^{2\phi(N)-d} \bmod N)$ 을 계산하여 사용하게 된다.

3.1 Rivain의 이중 멱승에 기반한 암호 시스템

이중 멱승 연산에서 각각의 지수 값 a와 b의 덧셈 체인 w는 [그림 2]와 같이 계산된다. 이 알고리즘을 계산하기 위해서 체인의 길이 u^*은 $E(u^*) \approx 2.03u$ 로 설정하는데 u값은 입력 값 a와 b의 비트 길이이다. 덧셈 체인 생성 알고리듬에 따라 체인 w를 계산한 후 메시지 M에 대한 이중 모듈러 멱승 연산을 수행하는 과정은 [그림 3]과 같이 나타낼 수 있다. 이와 같은 덧셈 체인을 이용한 이중 모듈러 멱승 연산에 기반한 RSA-CRT 구현 과정은 [그림 4]와 같다.

Double addition chain computation : Chain-Compute(a, b) 입력 : a, b $(a \le b)$ 출력 : w
1. $R_0 = a$, $R_1 = b$, $\gamma = 1$, $j = u^*$ 2. while$((R_{\gamma\oplus 1}, R_\gamma) \neq (0,1))$ do { 3. if$(R_\gamma/2 > R_{\gamma\oplus 1})$ 4. then $w_{j-1} = 0$, $w_j = R_\gamma \bmod 2$, $R_\gamma = R_\gamma/2$, $j = j-2$ 5. else $w_j = 1$, $R_\gamma = R_\gamma - R_{\gamma\oplus 1}$, $\gamma = \gamma\oplus 1$, $j = j-1$ } 6. Return(w)

[그림 2] 지수 쌍(a, b)에 대한 이중 덧셈 체인 알고리듬

[Fig. 2] Double addition chain algorithm for the pair (a, b)

Double exponentiation : Double-Exp(M, w, N)
입력 : M, w, N 출력 : $M^a \bmod N, M^b \bmod N$
1. $R_0 = 1$, $R_1 = M$, $\gamma = 1$ 2. for $i = 0$ up to u^* { 2.1 if $(w_i \equiv 0)$ then $\{R_\gamma = R_\gamma^2 \bmod N$, $i \leftarrow i+1$ 2.2 if $(w_i \equiv 1)$ then $R_\gamma = R_\gamma \cdot M \bmod N$ } 2.3 else $R_{\gamma\oplus 1} = R_{\gamma\oplus 1} \cdot R_\gamma \bmod N$, $\gamma = \gamma\oplus 1\}$ 3. Return$(R_{\gamma\oplus 1}, R_\gamma)$

[그림 3] 이중 모듈라 멱승 알고리듬

[Fig. 3] Double modular exponentiation algorithm

[그림 4]에서 보는 바와 같이 RSA-CRT 연산을 위해서 d_p와 $2(p-1)-d_p$ 의 덧셈 체인과 d_q와 $2(q-1)-d_q$ 의 덧셈 체인을 각각 구한다. 이와 같이 각각의 덧셈 체인으로 생성된 (w_p, w_q) 를 기반으로 (S_p, C_p) 및 (S_q, C_q)를 계산한 후 재결합 과정을 통해 최종 서명 S을 생성한다. 여기서 $S_p = M^{d_p} \bmod p$가 되고 $S_q = M^{d_q} \bmod q$가 된다. 마지막으로 [그림 4] 단계 6에서는 서명 S가 오류 없이 정상적으로 계산되었는지 확인하게 된다. 즉, 정상적으로 서명이 생성되면

$S \cdot C_p = M^{d_p} \cdot M^{2(p-1)-d_p} = M^{2\phi(p)} \bmod p = 1$및 $S \cdot C_q = M^{d_q} \cdot M^{2(q-1)-d_q} = M^{2\phi(q)} \bmod q = 1$이 성립하므로 오류를 검사하게 된다.

Secure RSA-CRT Algorithm : RSA-CRT(M, p, q, d_p, d_q, i_p) 입력 : M, p, q, d_p, d_q, i_p 출력 : $M^d \bmod N$
1. w_p = Chain-Compute(d_p, $2(p-1)-d_p$) 2. (S_p, C_p) =Double-Exp($M \bmod p$, w_p, p) 3. w_q = Chain-Compute(d_q, $2(q-1)-d_q$) 4. (S_q, C_q) =Double-Exp($M \bmod q$, w_q, q) 5. $S = CRT(S_p, S_q)$ 6. if ($S \cdot C_p \bmod p \neq 1$ or $S \cdot C_q \bmod q \neq 1$) then Return(Error) else Return(S)

[그림 4] 이중 멱승에 기반한 RSA-CRT 알고리듬

[Fig. 4] RSA-CRT algorithm based on the double exponentiation

3.2 Rivain의 이중 멱승 연산에 대한 취약성 및 효율성

이 절에서는 Rivain의 덧셈 체인을 이용한 이중 멱승 알고리즘의 취약점 및 효율성을 분석해 본다. Rivain의 이중 멱승 방식은 멱승 연산 과정에 더미 연산이 없어 C-Safe 오류 공격 방어할 수 있지만 기본적으로 DPA를 방어할 수 없다. DPA를 방어하기 위해서는 지수나 메시지 블라인딩과 같은 방법을 별도로 적용해야 한다. 하지만 지수 d를 랜덤 수 r을 이용하여 $d+r \cdot \phi(N)$으로 확장하여 구현한다면 DPA 공격은 방어할 수 있지만 전체 계산량은 지수의 길이만큼 더 늘어나는 단점이 있다.

특히, Rivain이 제안한 기법은 기본적으로 Bellcore 오류 주입 공격을 방어하기 위한 대응책으로 제안되었지만 보안 취약성을 가지고 있다. 이중 멱승에 기반한 RSA-CRT 알고리듬에서 모듈라 멱승을 시작하기 전 $M \bmod p$나 $M \bmod q$와 같은 연산을 하게 되는데 이 경우 오류를 주입하는 공격이 가능하다. 만약 [그림 4]의 단계 2나 단계 4에서 메시지 M에 대한 모듈라 연산 시 오류가 발생하면 단계 6의 검사 과정은 무사히 통과하게 된다. 예를 들어 단계 2에서 $M \bmod p$에 오류가 주입되어도 $S \cdot C_p \bmod p$값은 1이 되어 단계 6을 통과하여 오류 서명 $S' = m'^d \bmod N$을 출력하게 된다. 따라서 Rivain이 이중 멱승 기법은 메시지 입력에 대한 오류 주입에 의해 Bellcore 공격이 성립하고 모든 비밀 키가 노출될 수 있다.

Rivain은 이중 멱승 방식을 제안하면서 다른 구현 공격 방법에 비해 계산 효율성이 높은 점을 강조하였다. 일반적으로 구현 공격 대응책들은 l비트의 지수에 대해 약 $2l$번 정도의 모듈라 곱셈이 필요하지만 Rivain의 방식에서는 약 $1.65l$번의 모듈라 곱셈이 필요하다고 분석하였다. 그러나

지수 블라인딩을 이용한 DPA 방어 기법 등을 사용하면 이것보다 많은 모듈라 곱셈 연산이 필요하다. 또한, Rivain의 덧셈 체인을 이용한 이중 멱승 알고리즘에서는 암호 연산 과정에서 덧셈 체인을 계산하고 체인 값 w를 사전에 메모리에 저장하고 있어야 한다. 따라서 이 체인을 계산하는데 시간적인 계산 복잡도가 증가할 뿐만 아니라 비밀 키의 약 2배의 크기를 가지는 체인 값을 저장할 메모리가 추가로 필요하다.

4. 이중 멱승에 기반한 구현 공격 대응 알고리듬 제안

본 논문에서는 SPA를 비롯한 전력 분석 공격 뿐 아니라 오류 주입 공격 그리고 조합 공격까지 방어할 수 있는 새로운 이중 멱승 알고리듬을 제안하고자 한다. 또한, 이 알고리듬을 RSA-CRT 암호 시스템 구현에 적용하면 Bellcore 공격까지 대응할 수 있는 있도록 설계하고자 한다.

4.1 제안하는 이중 멱승 알고리듬

먼저 주어진 두 개의 지수 a와 b에 대한 이중 멱승 $(m^a \bmod N,\ \ m^b \bmod N)$ 쌍을 구하는 기본 알고리듬을 제안한다. 이 알고리듬은 기본적으로 SPA, Doubling 공격, Horizontal CPA, C-safe 오류 공격 그리고 조합 공격에 대응되도록 설계하였으며 이를 나타낸 것이 [그림 5]이다. 제안하는 알고리듬은 SPA 방어를 위해 정규적인 모듈라 곱셈이 되도록 하였으며 Doubling 공격에 강인하도록 Right-to-Left 형태로 구성하였다.

특히, 제안 알고리듬은 두 지수 a와 b의 각 비트에 따라 곱셈 결과를 저장하는 레지스터를 따로 사용하는 것이 특징이다. 즉, $a_i = 1$이고 $b_i = 0$이면 중간 계산 값을 $R[1]$ 레지스터에, $a_i = 0$이고 $b_i = 1$이면 $R[2]$ 레지스터에, $a_i = 1$이고 $b_i = 1$이면 $R[3]$ 레지스터에 그리고 a_i와 b_i가 동시에 0이면 $R[0]$ 레지스터에 각각 저장한다. 그리고 레지스터 R은 메시지에 대한 연속적인 자승 값을 계산하는 용도로 사용된다. 이런 연산 과정을 거치면 단계 3의 $T[1]$은 a_i가 1일 때의 R값을 저장한 결과가 되므로 $T[1] = M^a \bmod N$이 되고, 단계 4의 $T[2]$은 b_i가 1일 때의 R값을 저장한 결과가 되므로 $T[2] = M^b \bmod N$이 된다. 여기서 $R[0]$는 두 멱승의 결과 값과는 관련성이 없지만 SPA를 방어하기 위해 정규적 연산 값을 저장하는 더미 레지스터 역할을 하게 된다. 또한 반복문 내의 $R[t]$값과 R은 매번 갱신되므로 같은 피승수 값을 갖는 곱셈 연산은 존재하지 않게 되는데 이러한 특성으로 인해 Horizontal CPA 공격이 적용되지 않는다.

그러나 멱승 연산 과정에 더미 연산이 있다는 사실은 C-safe 오류 주입 공격에 취약할 수도 있다. 따라서 단계 5와 같은 오류 주입 검사 기법을 통하여 더미 연산에 오류가 주입되었는지 확인함으로써 C-safe 오류 주입 공격을 방어할 수 있도록 설계하였다. 단계 2에서 수행한 모든 반복 연산을 정상적으로 마치게 되면 다음과 같이 단계 5의 검사식이 성립함을 알 수 있다.

$$R = M^{2^l} \bmod N \tag{2}$$

$$R[0] \cdot R[1] \cdot R[2] \cdot R[3] = M^{2^l - 1} \bmod N \tag{3}$$

$$R = R[0] \cdot R[1] \cdot R[2] \cdot R[3] \cdot M \bmod N \tag{4}$$

$$\begin{aligned} R \cdot R[3] &= R[0] \cdot R[1] \cdot R[3] \cdot R[2] \cdot R[3] \cdot M \bmod N \\ &= R[0] \cdot T[1] \cdot T[2] \cdot M \bmod N \end{aligned} \tag{5}$$

한편, 이 알고리듬의 단계 1에 있는 $R[0] \sim R[3]$레지스터 중 하나가 0으로 초기화 되면 오류 주입 공격과 SPA를 결합한 조합 공격이 가능하다. 따라서 제안 알고리듬에서는 어느 특정한 레지스터 값이 0이 되지 않도록 하는 연산($R[0] = R[1] = R[2] = R[3] = (R[0] \wedge R[1] \wedge R[2] \wedge R[3])$)을 추가하여 조합 공격을 막을 수 있도록 설계하였다. 결론적으로 [그림 5]의 알고리듬은 상기한 구현 공격 중에서 SPA, Doubling 공격, Horizontal CPA, C-safe 오류 주입 공격 그리고 조합 공격에 대응하면서 두 개의 지수에 대한 멱승 연산을 수행할 수 있다.

Double Exponentiation Algorithm : New-Double-Exp(M, N, a, b) 입력 : M, N, a, b 출력 : ($M^a \bmod N$, $M^b \bmod N$) or Error
1. $R[0] = R[1] = R[2] = R[3] = 1$, $R = M$ $(R[0] = R[1] = R[2] = R[3] = (R[0] \wedge R[1] \wedge R[2] \wedge R[3]))$ 2. for $i = 0$ up to $l-1$ { 2.1 $t = a_i + 2b_i$ 2.2 $R[t] = R[t] \cdot R \bmod N$ 2.3 $R = R \cdot R \bmod N$ } 3. $T[1] = R[3] \cdot R[1] \bmod N$ 4. $T[2] = R[3] \cdot R[2] \bmod N$ 5. if($R \cdot R[3] \equiv R[0] \cdot T[1] \cdot T[2] \cdot M \bmod N$) then Return($T[1]$, $T[2]$) 6. else Return(Error)

[그림 5] 제안하는 이중 멱승 알고리듬

[Fig. 5] Proposed double exponentiation algorithm

다음 [그림 6]은 $a = 22$ 이고 $b = 19$ 일 때 제안하는 이중 멱승 알고리듬의 중간 계산 값들을 표시한 예로서 이 알고리듬을 통해 $T[1] = M^{22}$ 과 $T[2] = M^{19}$ 가 정확히 계산됨을 보여 주고 있다. 예제에서는 편의를 위해 모듈러 연산 표현은 생략하였다. 그림에서 보는 바와 같이 단계 2의 모든 반복문을 종료한 후에는 $R = R[0] \cdot R[1] \cdot R[2] \cdot R[3] \cdot M$관계가 성립함을 알 수 있는데 이 관계식이 멱승 연산 시 각 모듈라 곱셈 및 자승이 오류 없이 수행되었는지 판별하는 중요한 검사 조건이 된다.

그러나 [그림 5]에서 제안한 기본 이중 멱승 알고리듬은 DPA 공격에 대응할 수 없다. DPA 공격에 대응하기 위해서는 메시지 랜덤화 방법을 사용할 수 있는데 [그림 5]의 알고리듬에 메시지 블라인딩을 적용한 것이 [그림 7]이다. [그림 7]에서는 메시지 블라인딩을 위해 랜덤 수 r을 생성하고 그 역수 $r^{-1} \bmod N$을 계산하여 이 값을 레지스터의 초기 값으로 사용하였다. 이 경우 공격자는 반복문내의 중간 값들을 예측할 수 없기 때문에 DPA 공격을 적용할 수 없다. 각 레지스터에 $R[0] = R[3] = r^{-1} \bmod N$과 $R[1] = R[2] = r$과 같이 다르게 초기화시킨 이유는 반복문 이후 수행되는 단계 4와 5에서는 블라인딩을 쉽게 제거할 수 있도록 하기 위함이다.

Registers & Steps		t	$R[0]$	$R[1]$	$R[2]$	$R[3]$	R	$T[1]$	T[2]
Step 1			1	1	1	1	M		
Step 2	$i=0$	2			M		M^2		
	$i=1$	3				M^2	M^4		
	$i=2$	1		M^4			M^8		
	$i=3$	0	M^8				M^{16}		
	$i=4$	3				M^{18}	M^{32}		
Step 3								M^{22}	
Step 4									M^{19}

[그림 6] 제안하는 이중 멱승 연산 과정

[Fig. 6] Computation process of proposed double exponentiation

Double Exponentiation Algorithm : Double-Exp-Blind(M, N, a, b)
입력 : M, N, a, b
출력 : ($M^a \bmod N$, $M^b \bmod N$, $M^z \bmod N$, $M^c \bmod N$, $M^{2^l} \bmod N$)

1. Generate a random number r
2. $R[0] = R[3] = r^{-1} \bmod N$, $R[1] = R[2] = r$, $R = M$
   $(R[0] = R[3] = (R[0] \wedge R[3]),\ R[1] = R[2] = (R[1] \wedge R[2]))$
3. for $i = 0$ up to $l-1$ {
   3.1 $t = a_i + 2b_i$
   3.2 $R[t] = R[t] \cdot R \bmod N$
   3.3 $R = R \cdot R \bmod N$ }
4. $T[1] = R[3] \cdot R[1] \bmod N$
5. $T[2] = R[3] \cdot R[2] \bmod N$
6. Return($T[1]$, $T[2]$, $R[0]$, $R[3]$, R)

[그림 7] 메시지 블라인딩이 적용된 이중 멱승 알고리듬

[Fig. 7] Proposed double exponentiation algorithm with message blinding

제안하는 DPA 방어용 이중 멱승 알고리듬에서는 RSA-CRT 구현에서 Bellcore 공격과 지수 블라

인딩을 적용하기 위해 반환하는 값이 중요하다. 이중 멱승 결과를 나타내는 $T[1]$과 $T[2]$ 값뿐만 아니라 중간 계산 과정의 오류를 검사하기 위해 $R[0]$, $R[3]$ 그리고 R 값도 반환한다. 결론적으로 [그림 7]에 나타낸 이중 멱승 알고리듬은 블라인딩 기법이 적용된 것으로 SPA, Doubling 공격, Horizontal CPA, C-safe 오류 주입 공격, 조합 공격뿐만 아니라 DPA 공격도 방어할 수 있다.

4.2 이중 멱승을 이용한 RSA-CRT 알고리듬

본 절에서는 [그림 7]의 이중 멱승 기법을 적용한 RSA-CRT 연산 알고리듬을 제안하고자 한다. 상기한 Rivain의 RSA-CRT 연산 알고리듬에서는 모듈러스 p에 대한 멱승 $S_p = M^{dp} \bmod p$값을 계산하기 위하여 이중 멱승 연산 값 $(M^{d_p} \bmod p,\ M^{2(p-1)-d_p} \bmod p)$ 을 구하는 기법을 사용하였다. 그러나 [그림 7]에서 제안하는 이중 멱승 알고리듬에서는 두 지수의 덧셈 체인을 구하는 것이 아니기 때문에 이중 멱승의 두 지수를 블라인딩할 수 있다.

지수를 랜덤하게 블라인딩하는 목적은 크게 2가지이다. 하나는 DPA 공격을 방어할 목적이고 다른 하나는 조합 공격 시 사용하는 SPA를 무력화시키기 위해 사용한다. 조합 공격은 오류 주입 후 하나의 단일 파형을 관측하여 비밀 키를 찾아내는 공격이므로 제안 방식에서는 지수를 랜덤화하여 단일 파형 분석으로는 비밀 키를 추출할 수 없도록 설계하였다. 따라서 이중 멱승에 사용되는 두 지수를 블라인딩하기 위해 $(d_p - k)$와 k를 사용한다. 여기서 k는 랜덤한 정수 값이다.

이 경우 이중 멱승의 결과는 $(M^{d_p - k} \bmod p,\ M^k \bmod p)$가 될 것이고 이 두 값을 곱한 단계 4에서 $S_p = M^{d_p} \bmod p$을 구할 수 있다. 비슷한 방법으로 이중 멱승의 두 지수로 $(d_q - k)$와 k를 사용하면 이중 멱승의 결과는 $(M^{d_q - k} \bmod q,\ M^k \bmod q)$가 될 것이고 이 두 값을 곱하면 단계 7에서 $S_q = M^{d_q} \bmod q$을 구할 수 있다.

두 개의 멱승 값 S_p와 S_q가 구해지면 단계 8에서 이 두 값을 재결합하여 최종적인 서명 S를 계산하게 된다. 그리고 마지막 단계 9에서 서명에 대한 오류가 없는지 검사하게 된다. 다음과 같은 중간 값들의 관계식에 의해 오류 검사의 정확성을 증명할 수 있다.

$$\begin{aligned} S &= S_p \bmod p \\ &= T[1] \cdot T[2] \bmod p \\ &= (R[3] \cdot R[1]) \cdot (R[3] \cdot R[2]) \bmod p \end{aligned} \tag{6}$$

$$\begin{aligned} R_p \cdot R_p[3] &= R \cdot R[3] \bmod p \\ &= (R[0] \cdot R[1] \cdot R[2] \cdot R[3] \cdot M) \cdot R[3] \bmod p \\ &= R[0] \cdot (R[3] \cdot R[1] \cdot R[3] \cdot R[2]) \cdot M \bmod p \\ &= R_p[0] \cdot S \cdot M \bmod p \end{aligned} \tag{7}$$

위의 관계식은 모듈러스가 q인 연산에서도 동일하게 성립한다. 여기서 중요한 점은 [그림 8]의 단계 10에서 입력 메시지 M과 관련된 수식이 성립하는지 확인함으로써 단계 2나 단계 4의

$M \bmod p$나 $M \bmod q$에서 오류가 주입되었는지를 검사할 수 있다.

만약 단계 2나 4의 모듈라 감소 연산에서 오류가 있었다면 단계 9의 검사식이 성립하지 않게 되어 결국 오류 서명을 출력하지 않고 Error 메시지를 출력하게 된다. 단, 여기서 d_p나 d_q는 연산 과정에서 오류에 의해 그 값이 변경되지 않았음을 확인할 수 있어야 한다. 예를 들어, 단계 2에서 $a+k$가 다시 d_p와 같은지를 검사하여 변경 여부를 확인할 수 있다. 결론적으로 제안하는 RSA-CRT 알고리듬은 입력 메시지에 대한 오류 주입을 통한 Bellcore 공격도 충분히 방어할 수 있다.

Secure RSA-CRT Algorithm : RSA-CRT(M, p, q, d_p, d_q, i_p) 입력 : M, p, q, d_p, d_q, i_p 출력 : $M^d \bmod N$ or Error
1. Generate a random number k 2. $a = d_p - k$, $b = k$ 3. $(T_p[1], T_p[2], R_p[0], R_p[3], R_p)$= Double-Exp-Blind($M \bmod p$, p, a, b) 4. $S_p = T_p[1] \cdot T_p[2] \bmod p$ 5. $a = d_q - k$, $b = k$ 6. $(T_q[1], T_q[2], R_q[0], R_q[3], R_q)$= Double-Exp-Blind($M \bmod q$, q, a, b) 7. $S_q = T_q[1] \cdot T_q[2] \bmod q$ 8. $S = CRT(S_p, S_q)$ 9. if$((R_p \cdot R_p[3] \equiv R_p[0] \cdot S \cdot M \bmod p) \& (R_q \cdot R_q[3] \equiv R_q[0] \cdot S \cdot M \bmod q))$ then Return(S) 10. else Return(Error)

[그림 8] 지수 블라인딩 기반의 이중 멱승을 이용한 RSA-CRT 알고리듬

[Fig. 8] RSA-CRT algorithm using double exponentiation based on exponent blinding

4.3 제안하는 이중 멱승 알고리듬의 안전성 및 효율성 분석

본 절에서는 기존에 제시된 멱승 알고리듬과 논문에서 제안하는 [그림 7]의 이중 멱승 알고리듬을 안전성과 구현 효율성 측면에서 분석해 본다. 먼저 Square-and-Multiply-Always 멱승 알고리듬은 비밀 키 한 비트당 한 번의 반복문 연산을 수행하고 있으며 이 반목문내에서는 정규적으로 한 번의 모듈라 곱셈과 한 번의 자승을 수행한다. 따라서 이 알고리듬은 비밀 키와 연관되지 않은 정규적인 반복문 연산을 수행하므로 SPA 공격을 방어할 수 있다. 하지만 Square-Multiply Always 알고리듬은 더미 연산을 가지는 특징으로 인해 C-safe 오류 공격에 취약할 뿐만 아니라 Horizontal CPA 공격에 의해서도 비밀 키가 노출될 수 있다.

BHT 멱승 알고리듬은 SPA와 C-safe 오류 공격에 대응하면서 CRT-RSA 연산에서 오류 주입 공격을 방어하기 위해 제안되었던 BNP 알고리듬을 개선하여 DPA 공격도 방어할 수 있도록 개선한 것이다. 그러나 BNP 및 BHT 멱승 알고리듬은 특정 레지스터를 0으로 초기화시킨 후 단순 전력

분석을 수행하는 조합 공격에 취약하다는 것이 최근 밝혀진 상태이다[16].

Joye에 의해 제안된 Square-Multiply Ladder 방식[22, 23]은 많은 구현 공격에 견고한 특성을 보인다. 즉, 반복문의 매 루프마다 자승과 곱셈을 정규적으로 수행하는 특징으로 인해 SPA 공격을 방어할 수 있으며 Right-to-Left 형태의 비밀 키 탐색 구조로 인해 Doubling 공격에도 강인한 속성을 가진다. 그러나 Square-Multiply Ladder 알고리듬은 Horizontal CPA 공격에 의해 비밀 키를 노출시킬 수 있는 취약점을 가지고 있다.

이중 멱승 기법을 사용하는 Rivain의 방법은 RSA-CRT 연산의 오류 주입 공격을 방어할 것을 목적으로 제안된 것이다. 그럼에도 불구하고 메시지 M에 대한 오류 주입 공격에 의해 Bellcore 공격에 취약한 특성을 보인다. 뿐만 아니라 DPA 공격을 방어하기 위해서는 지수 블라인딩이나 메시지 블라인딩과 같은 별도의 대응책이 필요한 단점을 가지고 있다.

[표 1] 멱승 알고리듬에 대한 안전성 및 효율성 비교

[Table 1] Comparison on security and efficiency of exponentiation algorithms

Algorithm & Attack			**A. 1**	**A. 2**	**A. 3**	**A. 4**	**A. 5**
Security	Power Attack	SPA[2]	O	O	O	O	O
		DPA[7]	△	O	△	△	O
		Doubling[8]	O	O	O	O	O
		Horizontal CPA[10, 11]	X	O	X	O	O
	Fault Attack	C-safe Error[13]	X	O	O	O	O
		Bellcore(RSA-CRT)[3]	△	O	△	X	O
	Combined Attack	PACA[15]	O	X	O	O	O
Efficiency	Multiplications(ex. $l=1024$)		$2l$	$2l$	$2l$	$1.65l$	$2l+2$
	Additive computational load		α		α	$\alpha+\beta$	
	Inverse for DPA countermeasure		-	1	-	-	1
O : Secure, X : Not secure, △ : Need additive countermeasures α : Computational load for DPA countermeasure β : Computational load for generating an addition chain							
A. 1: Square-Multiply Always(L-to-R)[7] A. 2: BHT method(R-to-L)[21] A. 3: Joye's Square-Multiply Ladder(R-to-L)[22, 23] A. 4: Rivain's Method(R-to-L)[17] A. 5: Proposed Double Exponentiation(R-to-L)							

본 논문의 [그림 7]에서 제시한 DPA 방어용 이중 멱승 알고리듬은 기존의 구현 공격들을 모두 방어할 수 있도록 설계하였다. 먼저 멱승에 대한 SPA를 방어하기 위해 매 루프 연산 시 정규적으로 곱셈과 자승을 수행하도록 하였으며, DPA를 방어하기 위해 메시지 랜덤화 기법을 사용하였다. 또한, 비밀 비트 탐색을 위해서 Right-to-Left 구조를 사용함으로써 Doubling 공격을 방어하도록 하였으며 매 루프마다 사용되는 레지스터 값을 업데이트시킴으로써 Horizontal CPA에 대응하도록 설계하였다. 뿐만 아니라 [그림 7]의 이중 멱승 알고리듬을 표준 RSA 방식으로 멱승을 할 경우에

는 [그림 5]의 단계 5와 같은 검사 기법을 이용해 더미 연산의 오류 여부를 검사함으로써 C-safe 오류 주입 공격에 대응할 수 있다. 만약, RSA-CRT 연산 시에는 메시지 자체에 대한 오류 주입 여부까지도 검사할 수 있어 C-safe 오류 공격과 Bellcore 공격을 동시에 방어할 수 있다. 마지막으로 모든 레지스터 초기 값이 정확히 입력되도록 확인하는 기법을 사용하거나 RSA-CRT에서의 지수 블라인딩 기법을 사용함으로써 오류 주입 공격과 SPA을 결합한 조합 공격에 대응하도록 하였다.

상기한 멱승 알고리듬들을 연산 효율성 측면에서 비교해 볼 때, 대부분의 멱승 알고리듬들은 SPA 공격에 대응하기 위해 정규화된 알고리듬을 사용하므로 약 $2l$번 정도의 모듈라 곱셈을 수행한다. 따라서 DPA 공격을 위한 추가 연산을 제외하면 Rivain의 이중 멱승 방법이 가장 효율적인 방법으로 분석되고 있으며 약 $1.65l$번의 모듈라 곱셈이 필요하다. 그러나 이 알고리듬에서는 덧셈 체인을 계산하는 코드와 별도의 연산 시간(β)이 필요하다. 그리고 Square-Multiply Always, Square-Multiply Ladder, 그리고 Rivain의 알고리듬에서 DPA 공격 방어를 위해 지수 블라인딩이나 메시지 블라인딩과 같은 기법을 사용해야 한다면 여기에 필요한 연산(α)이 추가되어야 한다. 반면, BHT 알고리듬과 제안하는 이중 멱승 알고리듬에서는 DPA를 방어하기 위해 랜덤 수에 대한 역수를 구하는 연산이 한 번 필요하지만 역수 연산은 전체 멱승 연산량에 비해 무시할 수 있는 수준이다. 제안 멱승 알고리듬을 BHT 알고리듬과 계산 효율성 면에서 비교해 보면 거의 동일하다. 그렇지만 BHT 알고리듬이 취약한 조합 공격에 대응할 수 있다는 것이 큰 장점 중 하나이다.

5. 결론

본 논문에서는 RSA와 같은 공개 키 암호 시스템 구현에 사용되는 멱승 알고리듬을 정보보호용 디바이스에 장착하여 사용할 경우 발생할 수 있는 물리적 구현 공격에 대한 취약점 및 대응 알고리듬을 분석하였다. RSA-CRT 연산 시 사용될 수 있는 Bellcore 공격을 방어하기 위해 Rivain이 제시한 이중 멱승 알고리듬을 분석한 결과, 메시지 오류 주입 공격에 취약한 특성이 있으며 구현상 덧셈 체인을 구하거나 별도의 DPA 대응책을 세워야 하는 비효율적인 면이 있었다. 따라서 본 논문에서는 덧셈 체인을 사용하지 않으면서 기존의 구현 공격에 대응할 수 있는 이중 멱승 알고리듬을 제안하였다. 제안하는 이중 멱승 알고리듬은 RSA-CRT 연산에도 활용할 수 있으며 물리적 조합 공격이나 Bellcore 공격도 방어할 수 있도록 설계하였다. 결론적으로 제안하는 이중 멱승 알고리듬은 지금까지 제시된 주요 구현 공격에 대응할 수 있으며, 추가적인 시스템 파라미터 저장이나 계산 시간이 늘어나지 않아 제한적인 보안 디바이스 개발 환경에서도 효과적으로 사용할 수 있다.

References

[1] P. Kocher, Timing Attacks on Implementation of Diffie-Hellman, RSA, DSS, and Other Systems, CRYPTO'96, LNCS 1109, **(1996)**, pp. 104-113.

[2] P. Kocher, J. Jae, and B. Jun, Differential power analysis, CRYPTO'99, LNCS 1666, **(1999)**, pp. 388-397.

[3] D. Boneh, R. DeMillo, and R. Lipton, On the Importance of Checking Cryptographic Protocols for Faults, EUROCRYPTO'97, LNCS 1233, **(1997)**, pp. 37-51.

[4] National Institute of Standards and Technology, Advanced Encryption Standards, **(2001)**, FIPS PUB 197.

[5] R. Rivest, A Shamir, and L, Adelman, A method for obtaining digital signature and public-key cryptosystems, Comm. of the ACM, **(1978)**, Vol. 21 No. 2, pp. 120-126.

[6] D. Gordon, A survey of fast exponentiation methods, Journal of Algorithms, **(1998)**, Vol. 27, pp. 129-146.

[7] J. Coron, Resistance against differential power analysis for elliptic curve cryptosystems, CHES'99, LNCS 1717, **(1999)**, pp. 292-302.

[8] P. Fouque and F. Valette, The doubling attack- why upwards is better than downwards, CHES'03, LNCS 2779, **(2003)**, pp. 269-280.

[9] E. Brier, C. Clavier, and F. Olivier, Correlation power analysis with a leakage model, CHES'04, LNCS 3156, **(2004)**, pp. 135-152.

[10] M. Witteman, J. Woudenberg, and F. Menarini, Defeating RSA Multiply-Always and Message Blinding Countermeasures, CT-RSA'11, LNCS 6558, **(2011)**, pp. 77-88.

[11] B. Feix, M. Roussellet, and A. Venelli, Side-channel analysis on blinded regular scalar multiplications, Cryptology ePrint Archive, **(2014)**, Report 2014/191.

[12] M. Joye, J. Quisquater, F. Bao and R. H. Deng, RSA-type signatures in the presence of transient faults, Cryptography and Coding, LNCS 1355, **(1997)**, pp. 109-121.

[13] S. Yen, S. Kim, S. Lim, and S. Moon, A countermeasure against one physical cryptanalysis may benefit another attack, ICISC'01, LNCS 2288, **(2002)**, pp. 414-427.

[14] C. Couvreur and J. J. Quisquater, Fast decipherment algorithm for RSA public-key cryptosystem, Electronics Letters, **(1982)**, Vol. 18, pp. 905-907.

[15] F. Amiel, K. Villegas, B. Feix, and L. Mercel, Passive and Active Combined Attacks: Combining fault attacks and side channel analysis, FDTC'07, IEEE-CS, **(2007)**, pp. 92-102.

[16] H. Kim and J. Ha, A physical combined attack and its countermeasure on BNP exponentiation algorithm, Journal of The Korea Institute of Information Security & Cryptology(JKIISC), **(2013)**, Vol. 23, No. 4, pp. 585-591.

[17] M. Rivain, Securing RSA against fault analysis by double addition chain exponentiation, CT-RSA'09, LNCS 5473, **(2009)**, pp. 459-480.

[18] M. Joye and S. M. Yen, The Montgomery Powering Ladder, CHES'02, LNCS 2523, **(2002)**, pp. 291-302.

[19] S. Yen, L. Ko, S. Moon, and J. Ha, Relative doubling attack against Montgomery ladder, ICISC'05, LNCS 3935, **(2005)**, pp. 117-128.

[20] A. Boscher, R. Naciri, and E. Prouff, CRT-RSA Algorithm Protected Against Fault Attacks, WISTP'07, LNCS 4462, **(2007)**, pp. 237-252.

[21] A. Boscher, H. Handschuh, and E. Trichina, Blinded fault resistant exponentiation revisited, FDTC'09, IEEE-CS, **(2009)**, pp. 3-9.

[22] M. Joye, Highly regular right-to-left algorithms for scalar multiplication, CHES'07, LNCS 4727, **(2007)**, pp 135-147.

[23] M. Joye, Highly regular m-ary powering ladders, SAC'09, LNCS 5867, **(2009)**, pp 350-363.

