(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년12월23일
(11) 등록번호 10-0875461
(24) 등록일자 2008년12월16일

(51) Int. Cl.
H04L 9/28 (2006.01) *H04L 9/30* (2006.01)
(21) 출원번호 10-2008-0018039
(22) 출원일자 2008년02월27일
심사청구일자 2008년02월27일
(56) 선행기술조사문헌
WO2003050998 A1*
WO9808323 A1
US6081597 A
KR1020070011293 A
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
인하대학교 산학협력단
인천 남구 용현동 253 인하대학교
(72) 발명자
이문규
경기 부천시 원미구 상동 진달래마을 효성아파트 2226동 501호
(74) 대리인
특허법인태동

전체 청구항 수 : 총 5 항 심사관 : 임대식

(54) 전력 분석 공격 방지를 위한 엔티알유 다항식 컨볼루션연산 방법 및 컴퓨터로 읽을 수 있는 기록매체

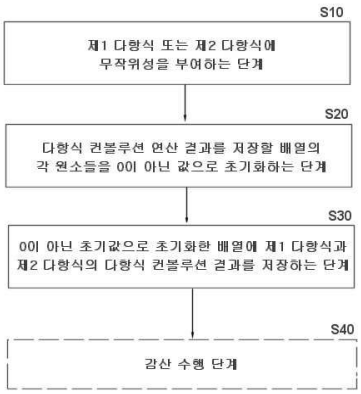
(57) 요 약

본 발명은 공개키 암호 체계인 NTRU 암호화 및 복호화에서 사용될 수 있는 다항식 컨볼루션 연산 방법 및 이를 기록한 컴퓨터로 읽을 수 있는 기록매체에 관한 것이다. 본 발명의 특징에 따른 다항식 컨볼루션 연산 방법은, NTRU 암호화 및 복호화에서, 제1 다항식('공개 키' 또는 '암호문'에 해당함)과 제2 다항식('개인 키' 또는 '임의 선택 다항식'에 해당함)과의 다항식 컨볼루션(convolution) 연산 방법으로서, (1) 상기 제1 다항식 또는 상기 제2 다항식에 무작위성(randomness)을 부여하는 단계; (2) 상기 제1 다항식과 상기 제2 다항식의 다항식 컨볼루션 연산 결과를 저장할 배열의 각 원소들을 0이 아닌 초기 값으로 초기화하는 단계; 및 (3) 상기 0이 아닌 초기 값으로 초기화된 상기 배열에, 상기 제1 다항식과 상기 제2 다항식의 다항식 컨볼루션 연산 결과를 저장하는 단계를 포함하는 것을 그 구성상의 특징으로 한다.

본 발명의 다항식 컨볼루션 연산 방법은, NTRU 암호 체계를 구성하는 중심 연산인 다항식 컨볼루션 연산을 전력 분석 공격에 대해 저항성을 가지도록 설계함으로써 안전한 암호화 시스템을 제공할 수 있으며, 특히 전력 분석 공격의 두 종류인 단순 전력 분석(simple power analysis: SPA) 및 차분 전력 분석(differential power analysis: DPA)을 모두 막을 수 있는 방법을 제공할 수 있다.

또한, 본 발명의 다항식 컨볼루션 연산 방법은, 큰 오버헤드 없이 전력 분석 공격을 방지하는 것에 의해, 다항식 컨볼루션 연산을 효율적으로 수행함으로써 보안 시스템의 성능을 일정 수준 이상으로 유지하는 방법을 제공할 수 있다.

대 표 도 - 도4



특허청구의 범위

청구항 1

NTRU 암호화 및 복호화에서, 제1 다항식('공개 키' 또는 '암호문'에 해당함)과 제2 다항식('개인 키' 또는 '임의선택 다항식'에 해당함)의 다항식 컨볼루션(convolution) 연산 방법으로서,

(1) 상기 제1 다항식 또는 상기 제2 다항식에 무작위성(randomness)을 부여하는 단계;

(2) 상기 제1 다항식과 상기 제2 다항식의 다항식 컨볼루션 연산 결과를 저장할 배열의 각 원소들을 0이 아닌 초기 값으로 초기화하는 단계; 및

(3) 상기 0이 아닌 초기 값으로 초기화된 상기 배열에, 상기 제1 다항식과 상기 제2 다항식의 다항식 컨볼루션 연산 결과를 저장하는 단계

를 포함하는 방법.

청구항 2

제1항에 있어서,

상기 단계 (1)는, 상기 제1 다항식의 원소들에 무작위수를 더함으로써 무작위성을 부여하는 것을 특징으로 하는 방법.

청구항 3

제1항에 있어서,

상기 단계 (1)는, 상기 제2 다항식에서의 계수 1의 위치를 나타내는 배열의 원소들의 순서를 무작위로 변경함으로써 무작위성을 부여하는 것을 특징으로 하는 방법.

청구항 4

제1항에 있어서,

상기 초기 값은 NTRU 암호화에서의 mod 연산의 피연산자인 q에 대한 배수 값을 갖는 것을 특징으로 하는 방법.

청구항 5

NTRU 암호화 및 복호화에서, 제1 다항식('공개 키' 또는 '암호문'에 해당함)과 제2 다항식('개인 키' 또는 '임의선택 다항식'에 해당함)의 다항식 컨볼루션(convolution) 연산 방법을 수행하는 컴퓨터 프로그램을 저장하기 위한 컴퓨터로 읽을 수 있는 기록매체에 있어서,

상기 다항식 컨볼루션 연산 방법은,

(1) 상기 제1 다항식 또는 상기 제2 다항식에 무작위성(randomness)을 부여하는 단계;

(2) 상기 제1 다항식과 상기 제2 다항식의 다항식 컨볼루션 연산 결과를 저장할 배열의 각 원소들을 0이 아닌 초기 값으로 초기화하는 단계; 및

(3) 상기 0이 아닌 초기 값으로 초기화된 상기 배열에, 상기 제1 다항식과 상기 제2 다항식의 다항식 컨볼루션 연산 결과를 저장하는 단계를 포함하는 것을 특징으로 하는 컴퓨터로 읽을 수 있는 기록매체.

명 세 서

발명의 상세한 설명

기 술 분 야

<1> 본 발명은 공개키 암호 체계인 NTRU 암호화 및 복호화에서 사용될 수 있는 다항식 컨볼루션 연산 방법에 관한 것으로서, 구체적으로는 보안 시스템의 전력 소모 패턴의 측정에 의해 내부의 비밀 키 정보를 알아내고자 하는 전력 분석 공격(Power Analysis Attack)에 대비하여, 연산의 단일화(unification) 및 무작위화(randomizatio

n)를 통해 안전한 보안 시스템을 구성하는 방법에 관한 것이다.

배 경 기 술

<2> 최근 인터넷 쇼핑, 인터넷 뱅킹, 은행카드 및 신용카드, 휴대전화 등 유무선 네트워크를 통해 개인의 중요한 정보들이 교환되면서 이를 보호하기 위한 수단의 중요성이 부각되고 있다. 따라서 개인 정보 보호를 위한 가장 중요한 이론적 기반인 암호 체계도 점차 더 많은 주목을 받고 있으며, 이미 30여 년 전부터 다양한 표준 기관에 의해 표준화가 진행되고 하드웨어 및 소프트웨어를 망라하는 다양한 방식으로 실현이 되어 왔다.

<3> 공개키 암호의 일종으로 최근에 개발된 NTRU 암호 체계는 매개변수의 선택에 따라 다양한 수준의 안전성을 제공할 수 있으며, 그 안전성의 수준을 기존에 알려져 있는 RSA 및 타원곡선암호와 같은 수준으로 조정할 때 이들보다 현저히 빠른 암호화 및 복호화 속도를 보여줌으로써 차세대 공개키 암호로서 새롭게 주목받고 있다. NTRU 암호 체계는 현재 IEEE에서 기존의 RSA 및 타원곡선암호 관련 표준인 IEEE P1363을 보완할 차세대 표준, 즉 IEEE P1363.1로 고려되고 있으며, 이미 표준화가 상당히 진행되어 2007년에는 Draft 9의 작업이 진행된 바 있다.

<4> 한편, 암호 체계의 안전성은 키(key)의 보호에 기인하므로, 암호 체계의 실현에 있어서는 키에 직접 관련된 연산은 시스템 내부에서 안전하게 보호된 상태로 수행하여 키 관련 정보를 외부로 유출하지 않도록 하는 것이 중요하다. 그러나 암호 체계를 규모가 작은 시스템, 예를 들어 스마트카드(smart card)나 내장형 시스템(embedded system) 등에 구현할 경우 그 구조의 단순성으로 인해 외부의 공격자가 이들 시스템의 전력 소모 패턴을 정밀하게 측정함으로써 키를 추측해 내는 것이 가능하며, 이를 전력 분석 공격(power analysis attack)이라 한다. 이 공격은 이론적으로 아무리 안전한 암호 체계를 설계한다 하더라도 그 구현 단계에서 물리적으로 암호를 무력화시키는 매우 강력한 공격 방법이다. RSA 및 타원곡선암호에 대해서는 전력 분석 공격을 막는 다양한 방법이 연구 개발되어 왔으나, NTRU 암호화에 대해서는 이러한 방법이 개발된 바 없다.

발명의 내용

해결 하고자하는 과제

<5> 본 발명은 기존에 제안된 방법들의 상기와 같은 문제점들을 해결하기 위해 제안된 것으로서, NTRU 암호 체계를 구성하는 중심 연산인 다항식 컨볼루션 연산을 전력 분석 공격에 대해 저항성을 가지도록 설계함으로써 안전한 암호화 시스템을 제공하며, 특히 전력 분석 공격의 두 종류인 단순 전력 분석(simple power analysis: SPA) 및 차분 전력 분석(differential power analysis: DPA)을 모두 막을 수 있는 방법을 제공하는 것을 그 목적으로 한다.

<6> 또한, 본 발명은 큰 오버헤드 없이 전력 분석 공격을 방지하는 것에 의해, 다항식 컨볼루션 연산을 효율적으로 수행함으로써 보안 시스템의 성능을 일정 수준 이상으로 유지하는 방법을 제공하는 것을 그 또 다른 목적으로 한다.

과제 해결수단

<7> 상기한 목적을 달성하기 위한 본 발명의 특징에 따른 NTRU 암호화 및 복호화를 위한 다항식 컨볼루션 연산 방법은,

<8> NTRU 암호화 및 복호화에서, 제1 다항식('공개 키' 또는 '암호문'에 해당함)과 제2 다항식('개인 키' 또는 '임의선택 다항식'에 해당함)의 다항식 컨볼루션(convolution) 연산 방법으로서,

<9> (1) 상기 제1 다항식 또는 상기 제2 다항식에 무작위성(randomness)을 부여하는 단계;

<10> (2) 상기 제1 다항식과 상기 제2 다항식의 다항식 컨볼루션 연산 결과를 저장할 배열의 각 원소들을 0이 아닌 초기 값으로 초기화하는 단계; 및

<11> (3) 상기 0이 아닌 초기 값으로 초기화된 상기 배열에, 상기 제1 다항식과 상기 제2 다항식의 다항식 컨볼루션 연산 결과를 저장하는 단계를 포함하는 것을 그 구성상의 특징으로 한다.

<12> 바람직하게는, 상기 단계 (1)는, 상기 제1 다항식의 원소들에 무작위수를 더함으로써 무작위성을 부여할 수 있다.

<13> 바람직하게는, 상기 단계 (1)는, 상기 제2 다항식에서의 계수 1의 위치를 나타내는 배열의 원소들의 순서를 무

작위로 변경함으로써 무작위성을 부여할 수 있다.

<14> 바람직하게는, 상기 초기 값은 NTRU 암호화에서의 mod 연산의 피연산자인 q에 대한 배수 값을 가질 수 있다.

<15> 본 발명의 또 다른 특징에 따른 NTRU 암호화 및 복호화를 위한 다항식 컨볼루션 연산 방법을 수행하는 컴퓨터 프로그램을 저장하기 위한 컴퓨터로 읽을 수 있는 기록매체는,

<16> 상기 다항식 컨볼루션 연산 방법이,

<17> (1) 상기 제1 다항식 또는 상기 제2 다항식에 무작위성(randomness)을 부여하는 단계;

<18> (2) 상기 제1 다항식과 상기 제2 다항식의 다항식 컨볼루션 연산 결과를 저장할 배열의 각 원소들을 0이 아닌 초기 값으로 초기화하는 단계; 및

<19> (3) 상기 0이 아닌 초기 값으로 초기화된 상기 배열에, 상기 제1 다항식과 상기 제2 다항식의 다항식 컨볼루션 연산 결과를 저장하는 단계를 포함할 수 있다.

효 과

<20> 본 발명의 다항식 컨볼루션 연산 방법은, NTRU 암호 체계를 구성하는 중심 연산인 다항식 컨볼루션 연산을 전력 분석 공격에 대해 저항성을 가지도록 설계함으로써 안전한 암호화 시스템을 제공할 수 있으며, 특히 전력 분석 공격의 두 종류인 단순 전력 분석(simple power analysis: SPA) 및 차분 전력 분석(differential power analysis: DPA)을 모두 막을 수 있는 방법을 제공할 수 있다.

<21> 또한, 본 발명의 다항식 컨볼루션 연산 방법은, 큰 오버헤드 없이 전력 분석 공격을 방지하는 것에 의해, 다항식 컨볼루션 연산을 효율적으로 수행함으로써 보안 시스템의 성능을 일정 수준 이상으로 유지하는 방법을 제공할 수 있다.

<22> 본 발명의 다항식 컨볼루션 연산 방법은, 신용카드, 현금카드 및 교통카드를 포함한 스마트카드, 인터넷 뱅킹에서 현재의 보안카드(난수표)를 대체할 차세대 보안 수단인 일회용 패스워드(one time password: OTP), WCDMA 등 3세대 이동통신에서 사용을 의무화하고 있는 범용 사용자 인증 모듈(universal subscriber identity module: USIM), 내장형 시스템(embedded system) 등에 대해 안전성을 제공할 수 있다.

<23> 또한, 본 발명의 다항식 컨볼루션 연산 방법은, 위에 나열된 것들 이외에도 전력 분석 공격이 가능한 모든 시스템에 대해 안전성을 제공할 수 있다.

발명의 실시를 위한 구체적인 내용

<24> 이하에서는 첨부된 도면들을 참조하여, 본 발명에 따른 실시예에 대하여 상세하게 설명하기로 한다.

<25> 먼저, 다항식 컨볼루션 연산, NTRU 공개-키 암호체계 및 전력 분석 공격에 대해 간단히 살펴본 후, 본 발명에 따른 다항식 컨볼루션 연산 방법을 살펴보기로 한다.

<26> 1. 다항식 컨볼루션 연산(polynomial convolution operation)

<27> Z를 정수들의 집합이라고 하자. Z[X]로 표시되는 Z에 대한 다항식 링은, Z의 계수들을 갖는 모든 다항식들의 집합이다. 몫 링(quotient ring) R을 $R = Z[X]/(X^N - 1)$ 로 정의하며, 이는 정수 계수를 갖는 임의의 다항식을 다항식 $X^N - 1$로 나누었을 때 생길 수 있는 모든 가능한 나머지 다항식들의 집합을 의미한다. 따라서 R에 속하는 원소 a는 다항식 또는 벡터로서 다음 수학식 1과 같이 적을 수 있다.

수학식 1

<28> $$a(X) = \sum_{i=0}^{N-1} a_i X^i = [a_0, a_1, \ldots, a_{N-1}]$$

<29> R에 속하는 원소 a와 b에 대한 컨볼루션 곱 c($c(X) = a(X) * b(X)$)는, 다음 수학식 2와 같이 표시되는 계수를 갖는다.

수학식 2

<30> $$c_k = \sum_{i=0}^{k} a_i b_{k-i} + \sum_{i=k+1}^{N-1} a_i b_{N+k-i} = \sum_{i+j \equiv k \pmod N} a_i b_j$$

<31> 여기서, $X^N \equiv 1 \bmod (X^N - 1)$ 이기 때문이다.

<32> 원칙적으로, 이 연산은 N^2개의 정수 곱셈을 필요로 하여 그 계산량이 많다. 그러나 NTRU에 의해 사용되는 다항식 컨볼루션 연산은, 일반적으로 a 또는 b 중 어느 하나가 작은 계수들을 가지며, 따라서 a*b의 계산은 매우 빠르게 수행될 수 있다.

<33> 2. NTRU 공개-키 암호체계

<34> 다양한 종류의 NTRU 암호체계가 존재하지만, Hoffstein이나 Bailey 등에 의해 제안된 개량 버전은 다음과 같이 설명될 수 있다:

<35> - NTRU는 3가지 공개 파라미터(N, p, q)를 가진다(여기서, p와 q의 최대공약수는 1이고, p ≪ q임).

<36> - 다항식의 계수들은 mod p or q로 감산된다(reduced mod p or q).

<37> - f^{-1} mod q로 표시되는 다항식 f의 mod q 상의 역원은, $f * f^{-1} \equiv 1 \bmod q$ 을 만족하는 다항식으로서 정의된다. 여기서, mod q는 다항식 내의 모든 계수가 mod q로 감산됨, 즉 q로 나눈 나머지로 계산됨을 의미한다.

<38> IEEE P1363.1 표준의 초안(working draft)은 NTRU에 대한 몇 가지 전형적인 파라미터 집합을 제안하는데, 그 중 하나는 (N, p, q) = (251, 2, 197)이다.

<39> 2.1 키 생성(Key Generation)

<40> R에 속하며 작은 계수들을 갖는 다항식 F, g를 임의로 선택한다. 그 후, $f := 1 + pF$ 와 $h := pf^{-1} * g \bmod q$ 을 계산한다. 여기서, mod q는 다항식 내의 모든 계수가 mod q로 감산됨, 즉 q로 나눈 나머지로 계산됨을 의미한다. 개인 키(private key)는 다항식 f이며, 공개 키는 다항식 h이다.

<41> 2.2 암호화(Encryption)

<42> m을 메시지를 나타내는 다항식이라고 하자. 작은 계수를 갖는 N-1차의 다항식 r을 임의로 선택하고, 암호문(ciphertext) $e := r * h + m \bmod q$ 을 계산한다.

<43> 2.3 복호화(Decryption)

<44> e를 복호화하기 위해 먼저 $a := e * f \bmod q$ 을 계산하되, a의 계수들이 $A \le a_i < A + q$를 만족하도록 선택한다. A의 값은 고정되며, 나머지 파라미터에 의존하는 간단한 공식에 의해 결정된다. 그 후, $m := a \bmod p$ 로서 평문(plaintext) m을 복구한다.

<45> 2.4 복호화의 유효성

<46> 2.3에서 계산된 다항식 a는 다음 수학식 3을 만족한다.

수학식 3

<47>
$$\begin{aligned} a &\equiv e * f \bmod q \\ &\equiv (r * h + m) * f \bmod q \qquad (\text{since } e \equiv r * h + m) \\ &\equiv pr * g + m * f \bmod q \quad (\text{since } h * f \equiv pg * f^{-1} * f \equiv pg) \end{aligned}$$

<48> 최종 다항식 pr*g + m*f를 고려해 보자. 파라미터들을 적당히 선택하는 것에 의해, q보다 작은 길이의 인터벌 내에 놓이도록 계수들을 조정할 수 있다. 따라서 우리는 다음 수학식 4와 같이 a를 복구할 수 있다.

수학식 4

<49> $$a = pr * g + m * f = pr * g + m * (1 + pF)$$

<50> 즉, mod q에 대해서가 아니라 정확한 등식이 성립하도록 할 수 있으므로 m = a mod p이다.

<51> 3. 전력 분석 공격 (Power Analysis Attacks)

<52> 도 1은 NTRU 암호화 시스템에 있어서 중심적인 연산인 다항식 컨볼루션 연산을 구현한 알고리즘의 일예를 나타내는 도면이다. 이 알고리즘은 0보다 크거나 같고 자연수 q보다 작은 정수를 계수로 갖는 일반 다항식 c(X)와, 이진수 1 또는 0을 계수로 갖는 이진 다항식 a(X)를 입력으로 하여 이들 두 다항식의 컨볼루션 곱인 t(X) = a(X)*c(X)를 계산한다. 이때 c(X)를 제1 다항식, a(X)를 제2 다항식이라 한다. 단, a(X)는 계수 1의 위치를 나타내는 배열인 b로 대체되어 입력되며, a(X) 및 c(X)의 계수의 총 개수는 N(정수)개이다. 또한 d는 b 배열 상의 원소의 개수이다.

<53> 여기서, 이진 다항식 a(X)는 복호화에서의 개인 키(private key) 또는 암호화에서의 임의 선택 다항식 r에 해당하며, 이 부분은 내부 연산에 해당하므로 외부에서 직접 접근할 수 없다. 즉, N개의 계수를 갖는 이진 다항식인 a(X)의 계수들 중 어느 것들이 1인지를 나타내는 배열 b는 외부에서 알 수 없어야 하며, 다만 일반 다항식 c(X)는 공개된 부분으로서 외부에서 접근이 가능할 뿐 아니라 공격자가 임의로 변조할 수도 있다. 공격자는 자신이 임의로 생성 또는 변조한 데이터를 입력으로 주어 이를 이용한 연산을 수행시킨 후 시스템 내부의 전력 소모 패턴을 외부에서 분석하여 간접적으로 키 정보를 유추해 내게 된다.

<54> (1) 단순 전력 분석(Simple Power Analysis: SPA) 공격

<55> NTRU에 대한 단순 전력 분석 공격에서 공격자는 시스템 내부에서 일어나는 덧셈 x + y 연산에 대해 x, y 중 어느 하나가 0인 경우와 x, y 모두 0이 아닌 경우에 대해 전력 소모의 미세한 차이를 감지할 수 있다. 이 사실을 이용하여 공격자는 일반 다항식 c(X)의 계수들을 0이 아닌 것들로만 구성한 후 시스템으로 하여금 다항식 컨볼루션을 수행하도록 하여 키 정보를 분석한다.

<56> 좀 더 구체적으로, 도 1에 따르면 NTRU 다항식 컨볼루션 알고리즘의 제5-7행은 두 정수의 덧셈을 N회 수행하는 것인데, $t_{k+b[j]}$ 값들은 처음에 모두 0으로 초기화되어 있으므로 제4행에서 j = 0으로 설정하였을 때는 제5-7행의 N개 연산은 모두 하나의 피연산자가 0인 덧셈이다. 그러나 다음 단계로 제4행에서 j = 1로 설정하였을 때는 N개 연산 중 일부는 피연산자가 모두 0이 아닌 덧셈이며, 나머지 일부는 피연산자 중 하나가 0인 덧셈이다. 후자의 경우가 어느 부분에서 발생하는지를 측정하면 b 배열의 첫 번째 원소인 b[0]과 두 번째 원소인 b[1]의 상대적인 차이 값을 구할 수 있다. 공격자가 이를 계속 반복하면 b 배열의 인접한 원소들 간의 상대적 차이 값을 모두 구할 수 있으며, 간단한 전수 조사(exhaustive search)에 의해 b[0]값만 구하면 b 배열 내의 모든 원소를 알아내어 키를 복원할 수 있다.

<57> 예를 들어, 다음은 N=7, b = [1, 4, 6]인 경우에 있어서, 도 1의 알고리즘의 제5-7행 부분에 대한 전력 소모 패턴을 시간 순으로 모식적으로 나타낸 것이다.

<58> j = 0: ZN ZN ZN ZN ZN ZN ZN

<59> j = 1: NN NN NN NN ZN ZN ZN

<60> j = 2: NN NN NN NN NN ZN ZN

<61> 여기서 ZN는 0과 0이 아닌 수 사이의 덧셈을 나타내며, NN은 0이 아닌 수와 0이 아닌 수 사이의 덧셈을 나타낸다.

<62> 상기 결과를 분석해 보면, j = 1일 때 후반부의 3개의 ZN 부분은 b[0]과 b[1]의 상대적인 차이가 3임을 의미하며, j = 2일 때 후반부의 2개의 ZN 부분은 b[1]과 b[2]의 상대적인 차이가 2임을 의미한다. 공격자는 b의 내용을 모르는 상태에서 위의 전력 소모 패턴만으로 b를 추측하게 되는데, b[0]을 x라 추정하면, b = [x, x+3, x+5]를 얻을 수 있다. 따라서 b의 원소 각각은 모르지만 실제로는 x 값 하나만을 구하면 b의 모든 원소들을 구할 수 있으므로 x 값을 전수조사하면 b를 알아낼 수 있다.

<63> 상기 공격 방법은 도 1의 알고리즘뿐 아니라 이와 유사한 형태를 가지는 모든 컨볼루션 알고리즘에 적용 가능한 방법이다.

<64> (2) 차분 전력 분석(Differential Power Analysis: DPA) 공격

<65> 단순 전력 분석 공격에서는 0과 0이 아닌 수 사이의 덧셈 및 0이 아닌 두 수 사이의 덧셈 간의 전력 소모 차이를 이용하였으나, 실제로는 0이 아닌 두 수 사이의 덧셈에서도 피연산자 값들의 차이에 의해 전력 소모에서 더욱 미세한 차이가 발생할 수 있다. 이러한 차이는 SPA와 같은 단순한 공격 방법으로는 감지하기 어려우나, 좀 더 고도화된 공격 방법인 차분 전력 분석(Differential Power Analysis: DPA)으로는 감지할 수 있다. NTRU에 대한 DPA에서 공격자는 일정한 이진 다항식 a(X)(따라서 일정한 배열 b)에 대하여 다양한 일반 다항식 c(X)들을 변화시키면서 적용하여 그 전력 소모 패턴을 측정한 후 그 결과를 통계적으로 분석하여 배열 b의 정보를 알아낸다. 따라서 공격자는 전력 소모 데이터를 SPA에서보다 더 많이 가지고 있으므로, 1회의 전력 분석에서는 감지할 수 없었던 미세한 차이를 누적시켜 더 큰 정밀도를 가지는 공격을 수행할 수 있다.

<66> 좀 더 구체적으로 설명하면, 공격자는 b 배열의 원소들을 알아내는 것이 목표인데, 도 1에 도시된 알고리즘의 제6행에서 $t_{k+b[j]} + c_k$의 결과 값이 최하위 비트가 0인 경우와 1인 경우에 대하여 공격자는 전력 소모의 미세한 차이점을 감지할 수 있다. 다만, 이 차이점은 SPA에서와 같이 1회의 전력 측정으로 구분해 낼 수 있는 차이가 아니므로, 다양한 c(X)에 대하여 연산을 수행하여 전력 소모의 차이를 누적시킴으로써 측정하게 된다. 이러한 공격이 가능한 것은, 같은 피연산자에 대한 덧셈은 같은 전력 소모 패턴을 가진다는 데에 기반하고 있다. 따라서 같은 피연산자와 관련된 계산이라도 항상 다른 전력 소모 패턴이 나오도록 함으로써 DPA를 방지하는 것이 가능하다.

<67> 다음으로 위에서 설명한 SPA 및 DPA 등 전력 분석 공격을 방지하기 위한 방법으로, 본 발명에 따른 다항식 컨볼루션 연산 방법을 상세히 살펴보기로 한다.

<68> 도 2는 본 발명의 일 실시예에 따른, 전력 분석 공격을 막기 위한 다항식 컨볼루션 방법의 일례를 구현한 알고리즘을 나타내는 도면이다. 본 알고리즘은 0보다 크거나 같고 자연수 q보다 작은 정수를 계수로 갖는 일반 다항식(제1 다항식) c(X)와, 이진수 1 또는 0을 계수로 갖는 이진 다항식(제2 다항식) a(X)를 입력으로 하여 이들 두 다항식의 컨볼루션 곱인 t(X) = a(X)*c(X)를 계산한다. 본 알고리즘은, (1) 상기 제1 다항식에 무작위성(randomness)을 부여하는 단계(제1-4행); (2) 상기 제1 다항식과 상기 제2 다항식의 다항식 컨볼루션 연산 결과를 저장할 배열의 각 원소들을 0이 아닌 초기 값으로 초기화하는 단계(제5-7행); 및 (3) 상기 0이 아닌 초기 값으로 초기화된 상기 배열에, 상기 제1 다항식과 상기 제2 다항식의 다항식 컨볼루션 연산 결과를 저장하는 단계(제8-12행)를 포함하며, 추가적으로 (4) 감산에 의한 보정 단계(제13-15행)를 더 포함한다.

<69> 상기 단계 (1)는 제1 다항식 c(X)의 원소들에 무작위수를 더함으로써 같은 입력에 대하여 알고리즘을 수행시켜도 전력 소모 패턴이 매번 다르게 나오도록 한다. 따라서 같은 입력은 같은 전력 소모를 발생시킨다고 가정하는 데서 기인하는 DPA 공격을 방지할 수 있다.

<70> 상기 단계 (2)는 컨볼루션 연산 결과를 저장할 배열 t의 각 원소를 0으로 초기화하는 대신 NTRU 암호화에서의 mod 연산의 피연산자인 q에 대한 배수로 설정함으로써 0과의 덧셈 자체가 일어나지 않게 하여 모든 덧셈이 유사한 전력 소모 패턴을 가지게 한다. 따라서 0에 대한 덧셈 판별에 기초한 SPA 공격을 방지할 수 있다.

<71> 상기 단계 (3)는 도 1의 제4-8행과 동일한 계산을 통하여 다항식 컨볼루션 연산 결과를 저장한다.

<72> 상기 단계 (4)는 단계 (1)에서 더해진 무작위수에 의한 오차를 보정하기 위해 감산을 수행한다. 이러한 보정이 가능한 이유는 b 배열 상의 원소의 개수 d가 항상 일정하다는 점 때문이다. 단계 (2)에서 배열 t의 원소를 0 대신 q로 초기화하는 데서 생기는 오차는 제 14행에서 mod q 연산을 함으로써 상쇄되며, 추가적인 보정이 필요하지 않다.

<73> 도 3은 본 발명의 또 다른 실시예에 따른, 전력 분석 공격을 막기 위한 다항식 컨볼루션 방법의 일례를 구현한 알고리즘을 나타내는 도면이다. 이 알고리즘은 (1) 제2 다항식에 무작위성(randomness)을 부여하는 단계(제1행); (2) 제1 다항식과 제2 다항식의 다항식 컨볼루션 연산 결과를 저장할 배열의 각 원소들을 0이 아닌 초기 값으로 초기화하는 단계(제2-4행); 및 (3) 상기 0이 아닌 초기 값으로 초기화된 상기 배열에, 상기 제1 다항식과 상기 제2 다항식의 다항식 컨볼루션 연산 결과를 저장하는 단계(제5-12행)를 포함하며, 감산에 의한 보정 단계는 포함하지 않는다.

<74> 상기 단계 (1)는 제2 다항식 a(X)의 계수 1의 위치를 나타내는 배열인 b의 원소들의 순서를 무작위로 변경함으로써, 같은 입력에 대하여 알고리즘을 수행시켜도 전력 소모 패턴이 매번 다르게 나오도록 한다. 따라서 같은 입력은 같은 전력 소모를 발생시킨다고 가정하는 데서 기인하는 DPA 공격을 방지할 수 있다. 배열 b의 원소의

순서를 변경하는 것은 연산의 결과에 영향을 미치지 않으므로 도 2의 알고리즘에서와 같은 보정 단계는 필요하지 않다.

<75> 상기 단계 (2)는 도 2에 도시된 알고리즘의 단계 (2)와 동일하므로, 이하 설명을 생략하도록 하겠다.

<76> 상기 단계 (3)는 도 1의 제4-11행과 동일한 계산을 통하여 다항식 컨볼루션 연산 결과를 저장한다.

<77> 도 4는 도 2에 도시된 알고리즘 및 도 3에 도시된 알고리즘을 하나의 플로 차트로 정리하여 나타내는 도면이다. 도 4에 도시된 바와 같이, 본 발명의 일 실시예에 따른 제1 다항식('공개 키' 또는 '암호문'에 해당함)과 제2 다항식('개인 키' 또는 '임의선택 다항식' 에 해당함)의 다항식 컨볼루션 연산 방법은, 제1 다항식 또는 제2 다항식에 무작위성을 부여하는 단계(S10), 제1 다항식과 제2 다항식의 다항식 컨볼루션 연산 결과를 저장할 배열의 각 원소들을 0이 아닌 초기 값으로 초기화하는 단계(S20), 및 (3) 0이 아닌 초기 값으로 초기화된 배열에, 제1 다항식과 제2 다항식의 다항식 컨볼루션 연산 결과를 저장하는 단계를 포함한다. 또한, 경우에 따라서, 본 발명의 일 실시예에 따른 다항식 컨볼루션 연산 방법은, 단계 S30 이후, 더해진 무작위수에 의한 오차를 보정하기 위해 저장된 다항식 컨볼루션 연산 결과로부터 감산을 수행하는 단계(S40)를 더 포함할 수 있다.

<78> 도 5 및 도 6은 전력 분석 공격에 대한 방지 기법이 적용되지 않은 도 1의 알고리즘을 이용하여 DPA 공격을 수행하는 실험의 예를 나타내는 도면이다. 도 5와 도 6의 가로축은 시간축이며, 세로축은 전력 신호의 크기를 나타낸다. 이 실험에서 공격자는 고정되어 있는 (그러나 공격자는 모르는) b 배열의 j번째 원소인 b[j]의 값을 임의로 추정한 후, 도 1에 도시된 알고리즘을 서로 다른 다수 개의 c(X) 다항식들에 대하여 수행하여 $t_{k+b[j]} + c_k$ 결과 값이 최하위 비트가 1인 경우와 0인 경우를 분류한 후 이들의 차이를 증폭시킨다. 추정이 잘못되었을 경우의 증폭 신호에서는 도 5와 같이 아무런 특징 없이 0을 중심으로 무작위 값들이 나타나게 된다. 만일 추정이 맞을 경우에는 도 6과 같이 일정 부분에서 신호의 극값(peak value)이 나타나게 된다. 이때 공격자는 b[j]의 여러 가능성에 대해 극값이 나타날 때까지 다른 값으로 재추정하여 분류 및 증폭하는 작업을 반복하면 된다.

<79> 본 발명의 특징에 따른 전력 분석 공격 방지 기법을 적용하면 도 6과 같이 극값이 나타나는 경우는 발생하지 않게 되며, 공격자가 정확한 추정을 하더라도 도 5와 유사한 증폭 그래프를 얻게 되어 자신이 추정한 값이 맞는 값인지 틀린 값인지 확인할 수 없어 공격에 성공할 수 없게 된다.

<80> 도 2 및 도 3의 전력 분석 공격 방지 알고리즘들은 제2 다항식 a(X)가 단순한 이진 다항식이 아니라 '승법형(product-form)' 의 다항식인 경우에도 유사하게 적용 가능하다. 여기서, 승법형(product-form)' 의 다항식이란, 이진 다항식 $a_1(X)$, $a_2(X)$, $a_3(X)$에 대하여 $a(X) = a_1(X)*a_2(X)$ 또는 $a(X) = (a_1(X)*a_2(X)+a_3(X))$ 또는 $a(X) = a_1(X)*a_2(X)*a_3(X)$와 같은 형태의 다항식을 말한다.

<81> 이상 설명한 본 발명은 본 발명이 속한 기술분야에서 통상의 지식을 가진 자에 의하여 다양한 변형이나 응용이 가능하며, 본 발명에 따른 기술적 사상의 범위는 아래의 특허청구범위에 의하여 정해져야 할 것이다.

도면의 간단한 설명

<82> 도 1은 NTRU 암호화 시스템에 있어서 중심적인 연산인 다항식 컨볼루션 연산을 구현한 알고리즘의 일예를 나타내는 도면.

<83> 도 2는 본 발명의 일 실시예에 따른, 전력 분석 공격을 막기 위한 다항식 컨볼루션 방법의 일예를 구현한 알고리즘을 나타내는 도면.

<84> 도 3은 본 발명의 또다른 실시예에 따른, 전력 분석 공격을 막기 위한 다항식 컨볼루션 방법의 일예를 구현한 알고리즘을 나타내는 도면.

<85> 도 4는 도 2에 도시된 알고리즘 및 도 3에 도시된 알고리즘을 하나의 플로 차트로 정리하여 나타내는 도면.

<86> 도 5 및 도 6은 전력 분석 공격에 대한 방지 기법이 적용되지 않은 도 1의 알고리즘을 이용하여 DPA 공격을 수행하는 실험의 예를 나타내는 도면.

<87> <도면 중 주요 부분에 대한 부호의 설명>

<88> c(X): '공개 키' 또는 '암호문'에 해당하는 제1 다항식

<89> a(X): '개인 키' 또는 '임의선택 다항식' 에 해당하는 제2 다항식

<90> t(X): 다항식 컨볼루션 연산 결과를 저장할 배열

<91> b: N개의 계수를 갖는 이진 다항식인 a(X)의 계수들 중 1의 위치를 나타내는 배열

<92> q: mod 연산의 피연산자

<93> S10: 제1 다항식 또는 제2 다항식에 무작위성을 부여하는 단계

<94> S20: 제1 다항식과 제2 다항식의 다항식 컨볼루션 연산 결과를 저장할 배열의 각 원소들을 0이 아닌 초기 값으로 초기화하는 단계

<95> S30: 0이 아닌 초기 값으로 초기화된 배열에, 제1 다항식과 제2 다항식의 다항식 컨볼루션 연산 결과를 저장하는 단계

<96> S40: 감산 수행 단계

도면

도면1

Input: b an array of d locations for '1' representing the polynomial $a(X)$; $c(X)$ the polynomial; N the number of coefficients in $a(X)$, $c(X)$.
Output: t the array where $t(X) = a(X) * c(X)$.

for $0 \leq j < 2N$ **do**
  $t_j \leftarrow 0$
end for
for $0 \leq j < d$ **do**
  for $0 \leq k < N$ **do**
    $t_{k+b[j]} \leftarrow t_{k+b[j]} + c_k$
  end for
end for
for $0 \leq j < N$ **do**
  $t_j \leftarrow (t_j + t_{j+N}) \bmod q$
end for

도면2

Input: b an array of d locations for '1' representing the polynomial $a(X)$; $c(X)$ the polynomial; N the number of coefficients in $a(X)$, $c(X)$.
Output: t the array where $t(X) = a(X) * c(X)$.

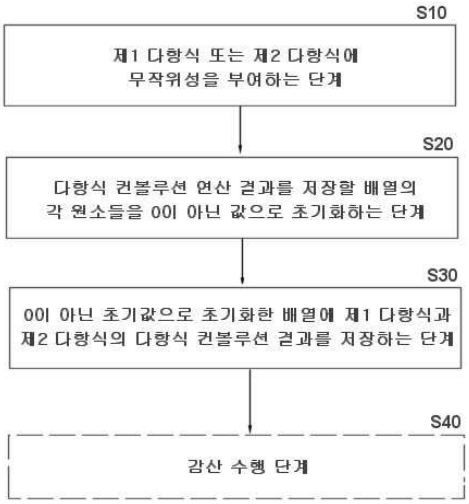
Generate a random r.
for $0 \leq j < N$ **do**
  $c_j \leftarrow c_j + r$
end for
for $0 \leq j < 2N$ **do**
  $t_j \leftarrow q$
end for
for $0 \leq j < d$ **do**
  for $0 \leq k < N$ **do**
    $t_{k+b[j]} \leftarrow t_{k+b[j]} + c_k$
  end for
end for
for $0 \leq j < N$ **do**
  $t_j \leftarrow (t_j + t_{j+N} - d \times r) \bmod q$
end for

도면3

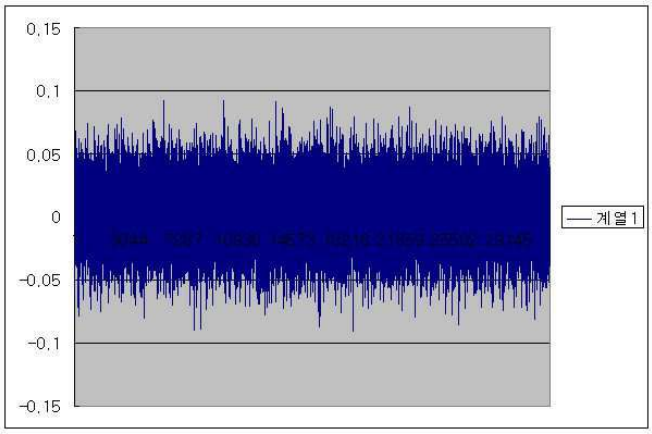
Input: b an array of d locations for '1' representing the polynomial $a(X)$; $c(X)$ the polynomial; N the number of coefficients in $a(X)$, $c(X)$.
Output: t the array where $t(X) = a(X) * c(X)$.

1: Randomly permute the elements in b.
2: **for** $0 \leq j < 2N$ **do**
3: $t_j \leftarrow q$
4: **end for**
5: **for** $0 \leq j < d$ **do**
6: **for** $0 \leq k < N$ **do**
7: $t_{k+b[j]} \leftarrow t_{k+b[j]} + c_k$
8: **end for**
9: **end for**
10: **for** $0 \leq j < N$ **do**
11: $t_j \leftarrow (t_j + t_{j+N}) \bmod q$
12: **end for**

도면4



도면5



도면6