정수계획법을 이용한 공개키 암호 알고리즘의 설계

용승림[U]· 조태남· 이상호
이 화 여 자 대 학 교 컴 퓨 터 학 과
{001COG03, tncho, shlee}@ewha.ac.kr

Design of a Public-Key Cryptographic Algorithm using Integer Programming

Seung-Lim Yong[U]· Tae-Nam Cho· Sang-Ho Lee
Dept. of Computer Science and Engineering, Ewha Womans University

요 약

공개키 암호 알고리즘의 암호화 함수는 한 방향으로의 계산은 매우 쉬우나, 역 계산은 매우 어렵다는 일방향성과 특별한 정보를 가지면 역 계산이 가능하다는 트랩도어(trapdoor) 성질이 있어야 하기 때문에 NP 문제나 계산상 풀기 어려운 수학 문제에 기반하여 연구되고 있다.

본 논문에서는 정수 계획법이라는 NP-완전 문제를 이용한 새로운 공개키 암호 알고리즘을 제안한다. 이 알고리즘의 키 생성 방식은 기존의 배낭꾸리기 암호 시스템의 방식과 유사하지만 기존 시스템의 공격 대상이었던 비밀키가 가지는 취약성을 보완하였다.

1. 서 론

공개키 암호 시스템은 현재 급속히 증가하고 있는 시스템 보안이나 컴퓨터 통신 보안 등에서 매우 중요한 기본 기술이다. 대부분의 공개키 암호 시스템은 수학분야의 풀기 어려운 문제나 계산 복잡도 이론상으로 풀기 어려운 문제를 기반으로 한다. 현재까지 수학분야의 풀기 어려운 문제를 기반으로 한 시스템들이 이용되고 있으며 아직까지 효율적인 공격법은 알려져 있지 않다. 그렇지만 계속적으로 시스템의 공격법에 대한 많은 연구가 이루어지고 있고, 시스템의 암호화와 복호화의 속도가 느리다는 단점이 있다. 계산 복잡도 이론상으로 풀기 어려운 문제인 NP 부류의 문제를 이용한 공개키 암호 알고리즘은 암호문의 길이가 평문의 길이보다 훨씬 더 커지거나 비밀키 설정상의 특성으로 인한 취약성에 대하여 많은 공격이 알려져 있다. 그러나 아직까지 많은 문제들이 공개키 암호 시스템에 이용될 수 있는 가능성을 가지고 있으며 그러한 문제들에 대한 암호학적 공격방법이 알려져 있지 않기 때문에 NP 부류의 문제는 암호학에서 중요한 의미를 갖는다.

본 논문에서는 정수 계획법이라는 NP 부류의 문제를 기반으로 한 새로운 공개키 암호 알고리즘을 제안한다. 이 알고리즘은 기존의 배낭꾸리기(knapsack) 암호 시스템의 방식과 유사하지만 비밀키의 특성으로 인한 취약성을 보완하였기 때문에, 이러한 취약성을 이용한 암호학적 공격 방법에 대해 안전하다. 또한 메시지를 암호화하고 복호화하는 시간이 짧아 효율적이고 시스템 구현이 간단하다는 장점이 있다.

2. 관 련 연 구

지금까지 제안되어 온 공개키 암호 시스템은 특별한 정보 없이 암호문을 복호화하는 것이 비효율적이도록 하기 위해서, 풀기 어렵다고 생각되는 수학 문제나 NP-완전 문제를 이용하여 개발되었다.

1978년 Ron Rivest, Adi Shamir 그리고 Leonard Adleman은 소인수 분해 문제의 어려움에 기반한 RSA라는 공개키 암호 시스템을 제안하였다[6]. 이후로 이산대수 문제의 어려움에 기반한 ElGamal의 공개키 암호 시스템과 합성수의 제곱근을 구하는 문제에 기반을 둔 Rabin의 알고리즘, 타원곡선 상의 이산대수 문제를 이용한 타원곡선 공개키 암호 알고리즘 등이 수학적 문제의 어려움에 기반한 공개키 암호시스템들이다 [1,2,5].

NP-완전 문제를 이용한 공개키 암호 시스템은 1978년 Merkle-Hellman의 배낭꾸리기(knapsack) 공개키 암호 시스템과 McEliece의 에러 수정 코드(error correcting code) 시스템 등이 있다[3,4]. 배낭꾸리기 암호 시스템은 NP 문제를 기반으로 한 최초의 공개키 암호 시스템으로서 NP-완전 문제로 알려진 부분합 문제를 기반으로 고안되었다. 그러나 초증가 수열의 특성이 약점이 되어 이에 대한 여러 가지 공격법이 제안되었고 제안된 공격법들에 의하여 안전하지 못함이 증명되었다.

3. 정수 계획법을 이용한 공개키 암호 알고리즘

정수 계획법이란 양의 정수 n, m과 $m \times n$ 크기의 정수 계수 행렬 A, 정수 계수의 벡터 B가 있다고 할 때 m개의 부등식 $A \cdot X \leq B$을 만족하는 정수 벡터 X가 존재하는지의 여부를 결정하는 문제로 NP-완전 문제로 알려져 있다. m개의 부등식 대신에 m개의 등식 $A \cdot X = B$을 만족하는 정수 계수 벡터 X가 존재하는지의 여부를 결정하는 문제 역시 NP-완전 문제임이 증명되어 있다[7].

정수 계획법을 이용한 공개키 암호 알고리즘은 공개키 암호 알고리즘이 만족해야 하는 일방향성과 트랩도어의 성질을 만족한다. 위 정의에서의 행렬 A를 공개키로, 암호화하고자 하는 평문을 X로, 두 행렬을 곱한 행렬 B를 암호문으로 이용한다. 행렬 A와 벡터 X를 알고 있을 때 $A \cdot X = B$를 계산하는 것은 쉽지만, 행렬 A와 B만을 알고 있을 때 벡터 X를 구하는 문제는 NP-완전 문제이므로 일방향성을 만족한다. 특별한 정보를 알고 있는 사람은 쉽게 해를 구할 수 있는 트랩도어 성질을 만족하기 위해서 모듈러 곱셈연산을 이용하여 NP-완전 문제를 일반적인 행렬의 곱셈문제로 변형한다. 정수 계획법을 이용한 공개키 암호 알고리즘의 각 단계별 상세 알고리즘은 다음절에서 기술한다.

3.1 키 생성

3.1.1 파라미터 설정 단계

공개키와 비밀키의 크리 파라미터 m과 n을 $n-m \leq m$의 조건을 만족하도록 선택한다. 파라미터 k는 메시지 블록의 크기를 결정해 주는 요소이다.

3.1.2 비밀키 생성 단계

비밀키는 0보다 큰 정수를 랜덤하게 선택하여 $(n-m) \times n$ 크기의 비밀키 행렬 S를 만들고 변수 w와 N을 다음과 같은 조건을 만족하도록 설정한다.

$$N : N > \max\left\{ \sum_{j=1}^{n} s_{ij} \cdot (2^k - 1) \mid i = 1, 2, \ldots, n-m \right\}$$

w : $1 < w < N$ 이고 $\gcd(w, N) = 1$ 인 정수

마지막으로, 공개키 생성시 행의 교환에 사용될 교환순열 π를 설정한다. 이와 같이 생성한 (S, w, N, π)가 비밀키가 된다.

3.1.3 공개키 생성 단계

공개키는 $m \times n$ 크기의 공개키 행렬 P로 비밀키를 변형하여 이루어진다. 처음 $n-m$ 행의 각 원소들은 S의 대응되는 원소값에 w를 곱하여 mod N 연산을 하고, 나머지 $2m-n$ 행은 N보다 작은 임의의 양의 정수로 선택한 후 각 행에 교환순열 π를 적용하여 만든다.

$P = (p_{ij})$,

$$p_{ij} = \begin{cases} w \cdot s_{\pi(i)j} \bmod N, & \text{if } 1 \leq i \leq n-m,\ 1 \leq j \leq n \\ r_{\pi(i)j} \in_R Z \bmod N, & \text{if } n-m < i \leq m,\ 1 \leq j \leq n \end{cases}$$

3.2 암호화

메시지를 암호화하기 위해서 평문을 일정한 길이의 블록으로 분할하고, 분할된 각 메시지 블록에 대해 암호화 알고리즘을 적용하여 각 메시지 블록에 대응하는 암호문 블록을 만든다.

평문 X에 대한 암호문 C는 평문의 각 블록을 암호화한 암호문 블록 $C_1, C_2, \cdots, C_u$의 집합으로 구성된다. 암호문 블록 C_i는 공개키 행렬 P에 메시지 블록 X_i를 곱하여 생성된 $m \times 1$ 행렬이다.

3.3 복호화

3.3.1 복호화 행렬 생성 단계

복호화 행렬은 $n \times n$ 정방 행렬 A로서 공개키 행렬 P와 비밀키 행렬 S로부터 생성된다. 교환 순열 π의 역순열 π^{-1}을 계산하고, 이를 행렬 P에 적용한 후 비밀키 행렬 S를 접합하여 $n \times n$ 정방 행렬 B를 만든다.

$$B = (b_{ij}), \quad b_{ij} = \begin{cases} p_{\pi^{-1}(i),j}, & \text{if } 1 \leq i \leq m,\ 1 \leq j \leq n \\ s_{i-m,j}, & \text{if } m < i \leq n,\ 1 \leq j \leq n \end{cases}$$

위의 방식으로 구한 행렬 B의 역행렬을 계산함으로써 복호화 행렬 A를 구한다.

3.3.2 평문 복호화 단계

공개키 행렬 P에 의해 암호화된 $m \times 1$ 크기의 암호문 블록들을 변형하고 확장하여 $n \times 1$ 크기의 확장된 암호문 블록 $C'_i = (c'_1, c'_2, \ldots, c'_m)$으로 만든다. 행렬 C'_i의 원소값 c'_1부터 c'_m은 C_i의 원소값에 역순열 π^{-1}을 적용하여 구하고, c'_{m+1}부터 c'_n은 각각 $c_{\pi^{-1}(1)}$ 부터 $c_{\pi^{-1}(n-m)}$ 값에 w^{-1}을 곱하고 mod N을 취하여 생성한다.

$$C' = (c'_i), \quad c'_i = \begin{cases} c_{\pi^{-1}(i)}, & \text{if } 1 \leq i \leq m \\ w^{-1} \cdot c_{\pi^{-1}(i-m)} \bmod N, & \text{if } m < i \leq n \end{cases}$$

평문 블록 X_i는 복호화 행렬 A에 확장된 $n \times 1$ 암호문 블록 C'_i를 곱함으로써 복원한다.

3.4 파라미터 및 변수의 설정 배경

비밀키 중 변수 N은 $N > \max\left\{ \sum_{j=1}^{n} s_{ij} \cdot (2^k - 1) \right\}$이어야 한다. 만일 이를 만족하지 않으면 확장된 암호문 블록 C'_i를 얻을 때, 서로 다른 두 개의 암호문 블록의 원소값에 대해 확장 암호문이 동일한 원소값 (즉, $c'_i = c'_j$ s.t. $c_i \neq c_j$)을 가지

게 된다.

또한 비밀키 (S, w, N, π) 중에서 w가 $1 < w < N$ 이고 $\gcd(w, N) = 1$인 조건을 만족하지 않을 경우, 공개키의 일부 원소가 되는 $w \cdot s_{ij} \bmod N$ 의 값이 0과 N-1 사이에서 고르게 나오지 않고 몇 개의 정수값에 치중된다.

4. 결과 및 분석

본 논문에서 제안한 알고리즘의 안전성과 효율성에 대하여 평가하고 배낭꾸리기 암호 시스템이 받은 공격에 대해 안전함을 보인다.

4.1 알고리즘의 안전성

$m \times n$ 공개키 행렬 P와 $m \times 1$ 암호문 블록 C_i을 이용하여 $P \cdot X_i = C_i$인 X_i를 구하는 것은 부등식 $m < n$을 만족하는 n개 미지수를 가지는 m개의 일차식으로 구성된 부정방정식을 푸는 문제와 같다. 메시지 블록의 크기를 k 비트라 하면 n개의 미지수 x_i와 m개의 식이 있을 때 조건을 만족하는 미지수의 값을 결정하기 위해서는 $2^{(n-m)k}$번 값을 대입해 보아야 한다. 따라서 대응하는 암호문에 대한 평문을 구하는 것은 지수 시간(exponential time)이 걸리게 된다.

배낭꾸리기 공개키 암호 시스템의 비밀키는 초증가 수열이며 공개키와 비밀키의 원소들은 일대일로 대응된다. Adi Shamir는 이러한 특성을 이용하여 배낭꾸리기 공개키 암호 시스템의 일반적인 공격법을 발표하였다[8,9]. 그러나 이러한 공격법은 비밀키가 초증가 수열이라는 특징을 가질 때 가능하다. 본 논문에서 제안한 공개키 암호 알고리즘은 비밀키의 원소값을 랜덤하게 선택하기 때문에 Shamir의 공격법은 본 알고리즘의 공격법이 될 수 없다.

4.2 알고리즘의 효율성

공개키 암호 알고리즘이 실용적이기 위해서는 암호화와 복호화가 빠른 시간에 이루어져야 한다. 암호화 단계에서 암호문 블록 C_i는 nm번의 곱셈과 nm번의 덧셈 연산을 수행해야 한다. 따라서 곱셈 연산과 덧셈 연산을 기본 연산으로 할 때 $O(nm)$의 시간 복잡도를 가진다.

복호화 단계에서는 복호화 행렬을 생성하고 암호문을 확장하여 복호화 행렬과 확장된 암호문을 곱한다. 복호화 행렬 A는 하나의 키 쌍에 대하여 한번만 계산하면 되므로, 시간 복잡도에는 큰 영향을 미치지 않는다. 확장된 암호문은 $2m - n$번의 곱셈 연산을 수행하고 복호화 하는데 곱셈과 덧셈 연산을 n^2번 수행하므로 $O(n^2)$의 시간 복잡도를 가진다. 따라서 암호화 단계나 복호화 단계 모두 다항식 시간 내에 수행될 수 있다.

5. 결론

본 논문에서는 정수계획법이라는 NP 부류의 문제를 이용한 새로운 공개키 암호 알고리즘을 제안하였다. 일반적인 전수조사로 암호문에 해당하는 평문을 얻기 위해서는 $2^{(n-m)k}$번의 지수 시간 연산을 필요로 한다. 공개키를 비밀키로 변형시키는 키 생성의 단계가 기존의 배낭꾸리기 공개키 암호 알고리즘과 유사하지만 배낭꾸리기 공개키 암호 알고리즘에서 보였던 비밀키의 취약점을 보완함으로써 유사한 암호학적 공격을 피할 수 있다. 또한 암호화와 복호화 단계에서의 시간 복잡도는 각각 $O(nm)$과 $O(n^2)$로서 다항식 시간에 암호화, 복호화할 수 있는 효율성을 보였다.

참 고 문 헌

[1] T. ElGamal, "A Public-Key Cryptosystem and A Signature Scheme Based on Discrete Logarithms," IEEE Transaction on Information Theory, Vol. IT-31, pp. 469-472, 1985.

[2] N. Koblitz, "Elliptic Curve Cryptosystems," Mathematic of Computation, Vol. 48, No. 177, pp. 203-209, 1987.

[3] R. J. McEliece, "A Public-Key Cryptosystem Based on Algebraic Coding Theory," Deep Space Network Progres Report, pp. 42-44, 1978.

[4] R. C. Merkle and M. E. Hellman, "Hiding Informatio and Signature in Trap-door Knapsacks," IEE Transactions on Information Theory, Vol. IT-24, pp 525-530, 1978.

[5] M. O. Rabin, "Digital Signatures and Public-Ke Functions as Intractable as Factorization," MI Laboratory for Computer Science, Technical Repo 1979.

[6] R. L. Rivest, A. Shamir, and L. Adleman, "A Method f Obtaining Digital Signatures and Public-k Cryptosystems," Comm. ACM, Vol. 21, pp. 120-126, 197

[7] S. Sahni, "Computationally Related Problems," SIAM Comput., Vol. 3, pp. 262-279, 1974.

[8] A. Shamir, "On the Cryptocomplexity of Knapsa System," Proc. 11th ACM Symp. on Theory Computing, pp. 118-129, 1979.

[9] A. Shamir, "A Polynomial Time Algorithm for Breaki the Basic Merkle-Hellman Cryptosystem," Proceedings the 23rd Annual Symposium on the Foundations Computer Science(IEEE), pp. 145-152, 1982.