수학원리와 특성 진단을 기반으로 한 공개키 RSA 알고리즘의 현장 적용 프로세스*

노 시 춘** · 송 은 지*** · 문 송 철****

목 차

요약

RSA 공개키 암호화 알고리즘에서는 소수, 키 생성, 소인수분해, 오일러 함수, 키 셋업, 합동식과 법, 지수 처리가 응용된다. 이와같은 알고리즘의 토대는 수학원리이다. 수학원리 중에서 첫 번째 개념은 소수를 구하여 응용하는 방법에서 출발한다. 두 개의 매우 큰 소수의 곱을 구하는 것은 용이 하지만 그 곱에서 원래의 두 개의 소수를 역 추적하는 것은 매우 어렵다는 원리를 이용한다. p와 q를 매우 큰 소수라 하면 이 두 개의 곱 n = p×q를 구하는 것은 쉽지만 역으로, 합성수인 n에서 p와 q를 추적하는 방법은 거의 불가능하다. RSA 암호화 알고리즘에서는 수학적으로 역함수 계산이 어려운 일방향 함수를 구현하기 위해 자리수가 많은 양의 정수의 소인수 분해 문제를 사용하고 있다. 역 방향으로의 계산을 어렵게 하기 위해 mod의 개념을 소인수 분해 문제에 더해서 사용한다. 암호화에 대한 관심분야는 대개 알고리즘 구현과 사용에 집중되고 있지만 막상 암호 알고리즘을 처음 도입하는 경우에는 어떤 프로세스를 거쳐야 현장 업무에 적용되는지를 알 수 없다. 본 연구는 공개키 알고리즘 속성 진단을 기반으로 한 현장 업무 암호화 적용 프로세스 방안을 제시한다.

표제어: RSA 암호 알고리즘, DES암호 알고리즘, 공개키, 비밀키, 수학 원리

접수일(2015년 7월 28일), 수정일(1차: 2015년 9월 23일), 게재확정일(2015년 9월 23일)
* 이 논문은 2014년도 남서울대학교 학술연구비 지원에 의해 연구되었음.
** 남서울대학교 컴퓨터학과 교수, nsc321@nsu.ac.kr
*** 남서울대학교 컴퓨터학과 교수, sej@nsu.ac.kr
**** 교신저자, 남서울대학교 컴퓨터학과 교수, moon@nsu.ac.kr

1. 서론

RSA 공개키 암호 알고리즘은 현재 공개키 암호화 에서 가장 널리 쓰이고 있는데 그 이유는 RSA 알고리즘이 최초로 공개키 암호화의 개념을 구현한 이유도 있지만, 그 안정성이 십여 년 이상을 통해 검증되었고 그 동안 발표되어 온 공개키 암호 알고리즘 중에서 이해와 구현이 쉽기 때문이다. RSA 기본 원리는 자리 수가 많은 양의 정수에 대한 소인수 분해가 어렵다는 것에 착안하여 이를 수학적으로 구현한 비대칭 알고리즘이다. RSA 공개키 암호화 알고리즘의 기본 형태는 합동식의 공식을 응용하는 DES 같은 대칭키 암호화와 같다. 대칭키 암호화와 다른 것은 암호화하는 키와 복호화 하는 키가 다르다는 것 밖에 없다. 이렇게 암호화 키와 복호화 키를 다르게 하기위해 RSA 암호화 알고리즘은 DES 같은 대칭 키 암호화 알고리즘과 확연히 다른 내부 알고리즘을 채용하고 있다. 암호 품질 속성은 이해관계자들의 관심사와 요구사항을 그대로 반영한다. 현장 적용 프로세스를 관점에서 보면 암호알고리즘은 이해당사자, 즉 관계자들이 원하는 수준으로 품질속성을 달성해야 한다. 본 연구 내용은 RSA 알고리즘의 수학원리 진단, RSA 알고리즘의 차별화된 특성 진단, 현장 적용 프로세스 제안 세 가지 측면으로 요약된다. 이 소재를 다루게 된 것은 암호화에 대한 관심분야는 대개 알고리즘 구현과 사용에 집중되고 있지만 막상 암호 알고리즘을 처음 도입하는 경우에는 어떤 프로세스를 거쳐야 현장 업무에 적용되는지를 알 수 없기 때문이다. 본 연구를 위해 RSA의 예제를 살펴보고, RSA 기법의 원리와 키 셋업, RSA 암호화 기법에서의 지수 연산 처리에 대해서 알아본다. 특히 RSA 알고리즘 현장 적용 방안을 도출 하여 사용의 참고 모델을 제시하는 데 목적을 두고 있다. 기술순서는 서론, 관련연구, RSA 알고리즘의 현장적용 프로세스 설계, 현장 적용 프로세스, 결론의 순서이다[2, 3].

2. 관련 연구

2.1 암호화 알고리즘 종류

2.1.1 대칭키 알고리즘 DES, 3DES(미국)

DES는 64비트의 평문을 46비트의 암호문으로 만드는 블록 암호 시스템으로 64비트의 키를 사용한다. 64비트의 키(외부 키) 중 56비트는 실제의 키(내부 키)가 되고 나머지 8비트는 거사용 비트로 사용한다. 또한 DES의 안전성을 증가시키기 위하여 키의 길이를 두 배 즉, 128비트, 십진수 16개를 키로 선택한 변형된 알고리즘을 일반적으로 사용한다. DES는 16라운드(Round)의 반복적인 암호화 과정을 갖고 있으며, 각 라운드마다 전치(Transposition) 및 대치(Substitution)의 과정을 거친 평문과 56비트의 내부 키에서 나온 48비트의 키가 섞여 암호문을 만든다. 복호화는 암호화 과정과 동일하나 사용되는 키만 역순으로 작용하는 것이다. 현재 DES는 안전하지 않다. DES는 컴퓨터 성능의 발달에 힘입어 보안성이 약화되어 2, 3DES를 사용하고 있다. 매 5년 마다 안전성을 검증하다 1997년에 NIST는 AES를 제시했고 2000년에 Rijndeal을 AES로 선택했다. 3DES(128bit key)는 기존의 DES 암호화 알고리즘방식을 다른 키에 세 번 적용시킨 것이며, 첫 번째 암호화과정, 두 번째 복호화과정, 세 번째는 또 다른 암호화 과정을 거친다.

2.1.2 공개키 알고리즘 ECC(미국)

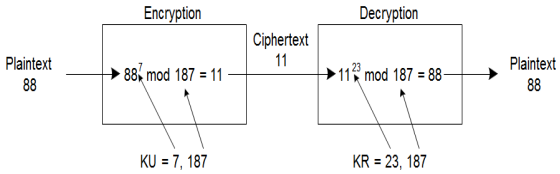
공개키 암호시스템에서의 타원곡선의 이용은 1985년에 Miller와 Koblitz가 독립적으로 제안하였다. 타원곡선 이론은 지난 100여 년간 연구되어온 대수기하의 한 분야로서 자연스러운 군연산과 그 연산을 수행해주는 효율적인 알고리즘을 가지고 있어 암호학적 응용이 용이하다. 다른 공개키 암호시스템에 비해 ECC가 가지는 장점들을 살펴보면 ① 주어진 소수 P에 대하여 유한체(FP)의 부분군을 이용하는 경우는 그 후보

가 유한체의 곱셈군(F×P) 밖에 없는 반면 ECC의 경우는 주어진 유한체상에서 정의된 다양한 타원곡선을 선택할 수 있어 풍부한 타원곡선군을 활용할 수 있다. ② 초 특이 곡선 같은 특별한 유형의 타원곡선을 제외하고는 알려진 이산대수 문제를 푸는 가장 효율적인 지수계산 알고리즘(index calculus algorithm)을 적용할 수 없어 안전한 암호시스템의 설계가 용이하다. ③ 다른 암호시스템에 비해 더 짧은 키 길이로 대등한 안전성을 주고 있어 훨씬 더 효율적이다. 예를 들어 RSA 1024비트 키와 ECC 160비트의 키는 대등한 안전도를 가진다. ④ 타원곡선상의 연산은 유한체의 연산을 포함하고 있으므로 H/W와 S/W로 구현하기가 용이하다. 모든 사용자가 동일한 유한체와 유한체 연산을 수행하는 같은 H/W를 사용하더라도 서로 다른 타원곡선을 선택하여 사용할 수 있으며 추가 보안을 위해 주기적으로 타원곡선을 바꿀 수 있다. ⑤ 비대칭키 방식의 암호화 알고리즘은 이러한 장점들 때문에 ECC는 스마트카드나 무선통신 단말기같이 메모리와 처리능력이 제한적인 응용분야에서 특히 각광받고 있다.

2.2 RSA 알고리즘의 수학원리

2.2.1 기본 원리

RSA 공개키 암호화 알고리즘에는 여러 가지 수학원리가 많이 사용된다. 첫 번째 개념은 소수를 구하여 응용하는 방법에서 출발한다. 두 개의 매우 큰 소수의 곱을 구하는 것은 용이하지만 그 곱에서 원래의 두 개의 소수를 구하는 것은 매우 어렵다는 원리를 이용한다. 즉 p와 q를 매우 큰 소수라 하면 이 두 개의 곱 n = p×q를 구하는 것은 쉽지만 역으로, 합성수인 n에서 p와 q를 추적하는 방법은 거의 불가능하다. RSA 암호화 알고리즘에서는 수학적으로 역함수 계산이 어려운 일방향 함수를 구현하기 위해 자리수가 많은 양의 정수의 소인수 분해 문제를 사용하고 있다. 역 방향으로의 계산을 어렵게 하기 위해 법의 개념을 소인수 분해 문제에 더해서 사용한다. RSA 공개키 암호화 알고리즘에서는 소수, 키 생성, 소인수분해, 오일러 함수, 키 셋업, 합동식과 법, 지수 처리가 시리얼하게 응용된다 [1, 4].



출처: JooSeok Song, The RSA Algorithm(2007).

그림 1. RSA 알고리즘 사례
Fig. 1. RSA Algorithm

2.2.2 소수

두 개의 매우 큰 소수의 곱을 구하기 위해 사용되는 소수는 1과 자신 이외에는 나누어지는 수가 없는 수를 말한다. 즉 3, 5, 7, 17, 19, 23, 27과 같은 수이다. RSA 암호의 아이디어는 중요 정보를 두 개의 소수로 표현한 후, 두 소수의 곱을 힌트와 함께 전송해 암호로 사용 하는 것. RSA 알고리즘에서는 모든 사람이 두 소수의 곱인 고유 한 n값을 갖는다. 만약 갑이 자신의 N을 p = 17,159와 q = 10,247의 곱인 N = 17,159×10,247 = 175,828,273으로 정하면 갑 자신의 N값을 모든 사람들에게 공개하면 이때의 N값은 갑의 공개키가 된다. 갑에게 메시지를 보내고 싶은 사람은 N값을 찾아 어떤 알고리즘을 통해 암호화를 한 후 갑에게 보낸다. 여기에서 p와 q는 갑의 사설 키이다.

2.2.3 오일러의 정리

오일러 파이 함수 φ(n)은 1부터 n-1까지의 양의 정수 중에서 n과 서로 소의 관계에 있는 정수들의 개수를 나타낸다. 두 개의 정수가 서로 소라 하는 것은 두 숫자 의 최대 공약수가 1인 것을 말 한다. 즉 1외에 는 두 숫자에서 공통적으로 나눌 수 있는 숫자가 없다. 오일러 파이 함수 φ(n)의 특별 한 경우로서, 다음이 성립한다. 만약 n이 소수라면, φ(n) = n-1이다.

또 양의 정수 n이 두 개의 소수 p와 q의 곱으로 이루어져 있다면, ϕ(n) = (p-1)×(q-1)이다. 예를 들면 ϕ(7)은 1부터 6까지 양의 정수 중 에서 7과 서로 소의 관계에 있는 정수들의 개수를 나타낸다. 7은 소수이므로, 위에서 설명한 공식을 사용하면 ϕ(7) = 7-1 = 6이 되며, 이는 위에서 실제로 구한 숫자들의 개수와 일치한다.

3. RSA 알고리즘의 현장적용 프로세스 설계

3.1 공개키 암호화 방향 정립

모든 공개키 암호화 알고리즘은 일 방향 함수를 사용하여 평문을 암호화한다. 일방 향 함수는 한쪽으로는 계산이 용이한 반면, 그 반대는 계산하기 어려운 함수이다. 한 쪽으로는 계산하기 쉬우므로 평문을 암호화하기는 쉽다. 그러나 반대쪽으로는 계산하기 어려워진다. 반대쪽으로 계산을 일방향 함수의 역 변환 이라한다. 만약, 자신을 포함한 모든 사람이 일방향 함수의 역변환을 할 수 없다면, 공개키 암호화에서는 암호의 기밀성은 보장되지만, 암호문을 복호화 해야 하는 자신도 암호문을 평문으로 바꿀 수 없다. 공개키 암호화에서는 두 가지 키가 제공 된다. 두 개 키 중 어떤 키라도 평문을 암호화 할 수 있다. 암호화 한 키는 일 방향 함수를 사용해서 평문을 암호화한다. 일방 향 함수의 특성상 반대쪽으로는 계산할 수 없으므로 암호문을 평문으로 바꿀 수는 없다. 그러나, 나머지 다른 하나의 키를 이용하면 이 역 변환을 쉽게 할 수 있다. 즉 나머지 키는 이런 반대쪽으로 변환하는 역변환 함수의 열쇠 역할을 하는 것이다.

3.2 암호화 요구사항 분석

요구사항은 암호화를 통해 얻을 수 있는 서비스 기능의 품질에 대한 요구조건이다.

3.2.1 소인수 분해 처리

RSA 암호의 기반이 되는 소인수분해는 임의의 소수를 이용하여 합성수를 만드는 것은 쉽지만, 합성수를 소인수분해 하여 소수로 만드는 것은 어렵다는 원리를 이용한다. 소인수분해 문제는 NP(비결정론적 다항식 시간)의 의미에서 어려운 문제라 알려져 있는데 예를 들면, 소수 23와 29의 곱으로 667이라는 합성수를 만들긴 쉽지만 667이라는 합성수를 보고 어떤 수의 곱으로 이루어졌는지 찾기 위해서는 시간이 필요하다. 특히, 작은 수가 아니고 300자리 이상의 아주 큰 소수의 곱으로 이루어진 RSA 시스템에서 그 소인수분해 과정은 시간적으로 장시간이 소요된다. 현장적용 시 이 원리를 이용하면 안정성이 매우 높아 널리 사용될 수 있다.

3.2.2 키 셋업 처리

각 유저는 공개키와 비밀 키를 가지고 있다. 우선 매우 큰 임의의 소수 p, q를 선택한다. 그리고 이 둘의 곱(n = pq)을 구한다. p와 q는 소수이기 때문에 오일러 토션 함수에 의해서 ø(n) = (p-1)(q-1)가 된다. 암호화를 위한 키 e를 선택한다. 이 때, 1 < e < ø(n), gcd(e, ø(n)) = 1를 만족해야 한다. 즉, ø(n)보다 작은 양수이며, ø(n)과도 서로소가 되어야 한다. 이제 복호화키 d를 구해야한다 이 때, ed = 1 mod ø(n) and 0 ≤ d ≤ n식을 만족하는 d를 구해야 한다. 그러면 키 셋업은 끝난다. 공개키는 {e, n}이 되며, 비밀키는 {d, p, q}가 된다.

3.2.3 합동식과 법(modulus) 응용

RSA 공개키 암호화 알고리즘의 융합된 수학원리 응용방법이 합동식이다. 합동식에서 법 또는 모듈러스(modulus)는 어떤 형식이 반복되는 경우에 그 반복되는 경계점을 의미한다. 모듈러 연산은 정수 연산의 한 종류이며 모든 숫자를 어떤 숫자 n에 대한 하나의 집합으로 단축이다. a ≡ b(mod m)의 식에서 정수 b를 법 m에 관한 a의 나머지라 하며, 역으로 a를 법 m에

관한 b의 나머지이다. 정수 a는 modulo m에 대해 b와 합동이다.

23 ≡ 11(mod 12)는 23은 법 12에 대해 11과 합동이다로 표현한다. '법'으로 하는 계산법에서 12 배수는 수치/값의 변화가 0과 같은 것으로 간주한다. 수치/값의 변화가 없다고 간주한다. 시계의 시간은 12시간 분으로 이루어져 있는데. 시간이 12시를 넘어가면 12시를 경계로 다시 처음인 1부터 시작이다. 이 12시를 법 또는 모듈러스라 한다. 시계의 초바늘은 12시인 법을 따른다. 법의 정의에 의하여 15시를 12시인 법을 다음과 같이 수학적으로 표현한다.

- 15 ≡ 3(mod 12)
- "15는법 12에 대해서 3과 합동이다."
- "≡" 기호는 합동식 부호이다.

합동식 부호는 동치를 나타내는 "=" 기호와 다르다. "≡"는 의미상 같다는 의미다. 12인 법에 대해서 15를을 연산한 값은 3임을 알기는 쉽다. 그러나 이와 반대인 상황인 12인 법에 대해서 연산한 값이 3일 때 원래의 값인 15를 찾기는 어렵다. 12인 법에 대해서 연산한 값이 3이 나올 수 있는 수는 15외에도 많다. 단지 3만으로 원래의 15이란 숫자를 찾는 것은 매우 어렵다. 따라서 법의 개념을 이용하면 단방향으로는 값을 찾기 쉬우나, 역방향으로는 찾기 어려운 함수를 만들 수 있다. 모듈러 연산의 특징은 다음과 같다.

[(a mod n)+(b mod n)] mod n = (a+b) mod n
[(a mod n)-(b mod n)] mod n = (a-b) mod n
[(a mod n)×(b mod n)] mod n = (a×b) mod n

3.2.4 지수 처리

RSA에서 모듈러 연산을 하기 위해, 지수 연산을 많이 하여 암호화, 복호화에 각각 쓰이게 된다. 이를 실수 연산을 한 뒤에 모듈러 연산을 하면 메모리와 연산 속도에서 현저히 떨어지게 된다. 이를 더 쉽고 빠르게 연산하기 위해서 알고리즘이 존재한다. 모듈러는 바이너리 값을 이용해서 연산을 하는 방식이다. ab mod n의 연산을 할 때, 알고리즘을 이용하면 간단하게 연산을 할 수 있다. 여기서 b는 바이너리 형태로 존재한다. 즉, b는 bkbk-1 … b0로 표현된다.

3.2.5 소수, 오일러의 정리, 합동식을 융합

ϕ(n)과 서로소의 관계에 있는 e를 구한다. 단 e는 1 < e < ϕ(n)의 범위에 있는 수이어야 한다. ϕ(n)과 e는 서로 소이므로 둘의 최대 공약수는 1이어야 한다.

Gcd(e, ϕ(n)) = 1

Gcd(a,b)는 a와 b 두 수의 최대 공약수를 나타낸다.

오일러의 정리는 법의 특수한 경우를 공식으로 만든 것이다. 두 양의 정수 a와 n가 서로 소라면 다음이 성립한다. aˆ ϕ(n) ≡ 1(mod n) 이것을 식으로 표현하면

표 1. 요구사항 분석항목
Tab. 1. Entry of Requirements Analysis

항목	기능설명	요구조건
소인수 분해 처리	작은 수가 아니고 300자리 이상의 아주 큰 소수의 곱으로 이루어진 RSA 시스템에서 그 소인수분해 과정은 시간적으로 장시간이 소요된다.	아주 큰 소수의 곱으로 이루어진 소인수 분해
지수 처리	모듈러 연산을 쉽고 빠르게 연산하기 위해서 모듈러는 바이너리 값을 이용해서 연산	바이너리 값을 이용해서 연산
소수, 오일러의 정리, 합동식 융합	aˆ ϕ(n) ≡ 1(mod n) → 3^4 ≡ 1(mod 5) → 81 ≡ 1(mod 5)로 표현될 수 있다. 이것을 식으로 고치면, → 81 mod 5 = 1이므로, 식의 계산이 맞다는 것을 알 수 있다.	모듈러 연산은 정수 연산의 한 종류이며 모든 숫자를 어떤 숫자 n에 대한 하나의 집합으로 단축
키 셋업 처리	우선 매우 큰 임의의 소수 p, q의 곱(n = pq)을 구한다. p와 q는 소수이기 때문에 오일러 토션 함수에 의해서 ø(n) = (p-1)(q-1)가 된다. 암호화를 위한 키 e를 선택한다.	1 < e < ø(n), gcd(e, ø(n)) = 1를 만족해야 한다.

a^ φ(n) mod n = 1이다. 이 공식이 맞는지 확인해보면 a = 3, n = 5로 해서 계산해 보면 5는 소수이므로 φ(5) = 5-1 = 4이다. 따라서, a^ φ(n) ≡ 1(mod n) → 3^4 ≡ 1(mod 5) → 81 ≡ 1(mod 5)로 표현될 수 있다. 이것을 식으로 고치면, → 81 mod 5 = 1이므로, 식의 계산이 맞다는 것을 알 수 있다. e×d ≡ 1(mod φ(n))의 식이 성립하는 d를 구한다. 즉 (e×d) mod φ(n) = 1이 되는 d를 구한다. 단 d는 d < φ(n)의 범위에 있는 정수이다. 개인키와 공개키의 값은 개인키: n, e 공개키: n, d이다.

4. 현장 적용 프로세스

4.1 도입 단계

4.1.1 암호 프로그램의 개발 또는 도입

인터넷 망에서 비밀정보를 암호화할 경우, 정부가 권장하는 암호 알고리즘을 적용한 제품을 활용하여 암호화한다. 내부 망에서 비밀정보를 암호화할 경우 자체에서 암호 프로그램을 개발하여 사용할 수 있다.

4.1.2 테스트 작업

실제 환경에서 테스트는 상당히 복잡한 측면이 있다. 알고리즘은 한 번 구축하고 나면 되돌리기 쉽지 않다. 따라서 알고리즘을 처음 선정할 때 반드시 테스트를 해 보는 것이 좋다. 테스트용 데이터 말고 개발 서버를 이용하여 실제 환경에서 테스트 해 보아야 한다. 어떤 데이터들을 암호화 하였고 색인은 어떻게 되어 있는지(암호화 하였는지 여부) 등에 이르기까지 세밀한 검토가 필요 하다.

4.2 실제 환경 테스트

4.2.1 환경 체크

작업에 필요한 여분의 Temporary Disk 공간은 있는지, 기타 DB의 환경변수들은 적절한지, 필요한 패키지나 패치 레벨 등이 적절히 설치되어 있는지 등을 체크한다. Encryption할 때마다 Size가 증가하여 암호화 처리 속도가 대칭키 방식에 비해 비교적 느리다. 대칭키 암호화방식과 공개키 암호화 방식의 대표적인 알고리즘인 DES와 RSA의 암호화 속도를 비교해 볼 때 DES가 약 1,000배 정도 빠른 것으로 나타난다. 그 원인은 알고리즘 자체가 수학에 기반을 두고 있어서 복잡한 계산의 과정을 거치기 때문이다.

4.2.2 완료 후 테스트

작업이 완료되고 나면 필요에 따라 암호화한 데이터가 정확히 변환 되었는지 정합성 체크를 실시한다. 애플리케이션을 수행하여 문제가 있지는 않은지 성능 저하는 어느 정도인지를 확인 한다.

4.3 운영 단계

4.3.1 암호 프로그램 관리자 지정

암호 프로그램을 관리하는 부서의 책임자는 암호 프로그램의 취급관리를 위해서 암호 프로그램 관리자를 업무별로 지정하고 암호관리방안을 정의한 암호 프로그램 관리방안 정의서를 작성하여 IT보안관리자의 승인을 득한다.

4.3.2 암호 프로그램의 운영

암호 프로그램을 변경할 경우에는 IT보안관리자의 승인을 득한다.

암호 프로그램의 경우 비밀자료에 준하여 관리하여야 한다.

4.4 현장 적용 영향평가

영향 분석은 암호화로 인한 기존 애플리케이션의 운영 에 어떤 영향이 있을 것 인지를 사전에 면밀히 분석 한다. 평가는 암호화가 가져올 수 있는 프로젝트의 리스크를 최소화한다. 적합성 평가 항목은 필요한 품질 속성을 달성하고 있는가. 기능성, 신뢰성, 사용성, 효율성을 다음 표의 기준으로 평가한다.

4.4.1 검증기간을 거친 안정성

- 큰 숫자의 소인수분해의 어려움을 이용하여 안전성을 높이고 있다.
- 모듈러(Mod) 연산을 통한 큰 숫자의 암호화 연산은 쉽지만, 그 역을 추적해하는 것은 매우 어렵기 때문에 RSA 방식의 안전도가 보장된다.
- RSA 알고리즘은 Key Management가 불필요하다.
- 수학원리를 사용하나 난이도는 높지 않은 편이며 경량화 방식의 장점을 가지므로 전자서명 기능과 인증(authentication)에 적합하다.

4.4.2 모듈러(Mod) 연산의 장점

- 암호화에서 Me mod n의 연산은 쉽다. 반대로 그 역 연산은 매우 어렵다. 큰 숫자의 암호화 연산은 쉬우나. 그 역을 추적해하는 것은 매우 어렵다.
- Mod 연산을 통해서 수 없이 많은 값 들이 M값이 될 수 있다.

4.4.3 블록단위 처리의 장점

- 블록단위로 암호화 하며, 각 블록은 n(키 값의 곱) 보다 작은 바이너리 값으로 이루어져 있다.
- 블록 사이즈는 log2(n)보다 작거나 같다. 암호화 방법은 C = Me mod n 방법이며 복호화는 M = Cd mod n = (Me)d mod n = Med mod n의 형태이다. 블록단위 처리는 강도에서 유리하며 단위 시간당 처리속도가 스트링 방식에 비해 상대적으로 유리하다.

4.4.4 사설키(Private Key)로 암호화

- 송신자와 수신자는 서로 n값을 알고 있으며, 공개 키는 {e, n}이며, 비밀키는 {d, n}이다.
- 공개키로 암호화 시스템은 공개키로 복호화할 수 없다.
- 사설키(Private Key)로 암호화한다는 것은 데이터의 기밀성 유지에 쓰이는 것이 아니라 인증과 무결성을 제공하기 위해 사용된다.

4.4.5 Message, 발신원 부인방지

- 송신자의 개인키로 암호화
- 송신자 이외에는 암호화할 수 없으므로 송신인에 대한 인증 및 송신 부인봉쇄가 가능하다. 단, 공개키를 소유한 자는 누구나 메시지를 복호화 할 수 있으므로 기밀성은 제공되지 않는다.

4.4.6 키관리 안전도

암호 알고리즘의 안전성은 데이터를 암호화한 후 암호 화 키와 함께 전송하기 때문에 키관리가 매우 중요하다. 고도의 암호화 알고리즘을 사용해 암호화한 데이터라 해도 암호화 키가 함께 유출되면 쉽게 복호화할 수 있기 때문이다. 평문의 각 문자 정보가 암호문 전체에 분산되는 특성을 확산(diffusion)이라 하며, 암호 해독자는 해독을 위해 더 많은 양의 암호문을 필요하게 된다. 또한 암호 해독자는 평문의 어떤 문자가 암호문에서 어떤 문자로 바뀌는지 알 수 없어야 한다.

4.4.7 테스트 편리성

암호화의 현장 적용을 위해서는 테스트가 사전에 수행되어야 한다. 이때 현장 테스트가 현장에서 운용환경에서 실시되는 것이 핵심이다. 현장 테스트는 데이터에 대한 암호화 프로세스와 품질의 체크가 관건이며 데이터베이스 암호화는 다른 보안 솔루션 적용과는 상당히 복잡한 측면이 있다. 데이터베이스 암호화는 한번 구축하고 나면 되돌리기 어렵기 때문에 처음 암호화 방법론 선택이 매우 중용하고 선정단계에서 테스트가 필요하다. 개발용 서버를 사용하여 실제 환경에서 테스트가 중용하며 어떤 데이터들을 암호화 하였고 색인은 어떻게 되어 있는지 까지 세밀한 검토가 필요하다.

표 2. 암호화 영향 평가 항목
Tab. 2. Encryption Impact Assessment Item

항목(factor)	세부항목(subfactor)	
시스템 기능성 (Functionality)	암호화 적합성 (Suitability)	특정 조건에서 암호화, 복호화 작업을 처리하는 기능의 적합성 에 영향을 미치는 속성
	시스템 상호운영성 (Interoperability)	특정 조건에서 암호화 기능과 시스템 기능간 상호 작용하는 속성
시스템 사용성 (Usability)	시스템 운영성 (Operability)	특정 상황에서 암호화 기능 수행 시 시스템 운영 관리에 필요한 사용자 노력에 영향을 미치는 속성
	키관리 안전도 (Key Management Safety)	특정 상황에서 암호화 기능 수행 시 암호 해독을 위해 더 많은 양의 암호문을 필요하게 되고 평문의 어떤 문자가 암호문에서 어떤 문자로 바뀌는지 알 수 없게 하는 속성
	암호화 테스트 편리성 (Usability Testing)	특정 상황에서 암호화 기능 수행 시 암호화의 현장 적용을 위해 테스트가 사전에 수행되어야 하고 현장 테스트가 현장 운용환경에 편리하게 진행되는 속성
시스템 효율성 (Efficiency)	시스템 효율성 (Time behavior)	특정 조건에서 암호화 기능을 수행할 때 적절한 시스템 응답시간, 처리효율, 처리시간을 제공할 수 있는 능력
	시스템 자원효율성 (Resource Behavior)	특정 조건에서 암호화 기능을 수행할 때 적절 양의 시스템 자원을 소모하는 능력
시스템 신뢰성 (Reliability)	시스템 오류허용성 (Fault Tolerance)	특정 상황에서 암호화 기능 수행 기간 동안 시스템 오류, 상호작용 방식에 문제 발생 시 정립된 프로세스가 동작하는 속성

5. 결 론

RSA 암호화 알고리즘은 현재 공개키 암호화에서 가장 널리 쓰이는 알고리즘이다. 널리 쓰이는 이유는 RSA 알고리즘이 최초로 공개키 암호화의 개념을 구현한 이유도 있지만, 그 안정성이 십여 년 이상을 통해 검증이 되었기 때문이다. RSA 공개키 암호화 알고리즘에는 여러 가지 수학원리가 많이 사용된다. 첫 번째 개념은 소수를 구하여 응용하는 방법에서 출발한다. 두 개의 매우 큰 소수의 곱을 구하는 것은 용이하지만 그 곱에서 원래의 두 개의 소수를 구하는 것은 매우 어렵다는 원리를 이용한다. 즉 p와 q를 매우 큰 소수라 하면 이 두 개의 곱 n = p×q를 구하는 것은 쉽지만 역으로, 합성수인 n에서 p와 q를 추적하는 방법은 거의 불가능하다는 원리이다. RSA 암호화 알고리즘에서는 수학적으로 역함수 계산이 어려운 일방향 함수를 구현하기 위해 자리수가 많은 양의 정수의 소인수 분해 문제를 사용하고 있다. 역 방향으로의 계산을 어렵게 하기 위해 법의 개념을 소인수 분해 문제에 더해서 사용한다. RSA 공개키 암호화 알고리즘에서는 소수, 키 생성, 소인수분해, 오일러 함수, 키 셋업, 합동식과 법, 지수처리가 시리얼하게 응용된다. Message 내용 또는 발신원에 대한 부인 방지 송신자의 개인키로 암호화하는 경우송신자 이외에는 암호화할 수 없으므로 송신인에 대한 인증 및 송신 부인봉쇄가 가능하나 공개키를 소유한 자는 누구나 메시지를 복호화 할 수 있으므로 기밀성은 제공되지 않는다. 본 연구에서 제안하는 공개키 알고리즘 속성 진단을 기초로 한 현장 업무 암호화 프로세스는 암호화를 업무에 처음 도입하는 현장에서 하나의 참고 모델이 될 수 있을 것으로 기대한다.

참 고 문 헌

〔국외 문헌〕

[1] Daniel Lloyd Calloway, Dr. Hannon(Instructor), (2008), “Literature Review of Cryptography and its Role in Network Security Principles and Practices”, Lecture Notes, Capella University, OM-

8302, 8 Sept.

[2] Sairam Natarajan #1 (2011), "A Novel Approach for Data Security Enhancement Using Multi Level Encryption scheme", Research paper, IJCSIT, 2 (1), 469-473.

[3] Himanshu, G. (2011), "Multiphase Encryption Technique", An Article, Amity University U.P., http://en.wikipedia.org/wiki/A_New_Concept_for_Multiphase_Encryption_Technique.

[4] Stallings, W. (1999), "Cryptography and Network Security: Principles and Practices", Prentice Hall.

[5] Walter, T. (1997), "A brief history of the data encryption standard", ACM Press/Addison-Wesley Publishing Co. NY, USA, 275-280.

[6] Sunita, B. and Prof. S. K. Sharma (2011), "Block Wise Parallel Encryption through Multithreading Concept", Research Paper published in Aishwarya Research Communication Journal(ISSN: 0975-3613), 3, 100-106.

[7] Noh, S. C. (2011), "A Study of Security QoS(Quality of Service) Measurement Methodology for Network Security Efficiency", Journal of Korea Convergence Security Association, 10(3), 40-41.

[8] Cho, B. K. and Jung, J. Il. (2010), "Policy-Based QoS Control Management System for VoIP Service", Journal of Korea Convergence Security Association, 10(3), 70-72.

[9] Wakefield, D. S., Mehr, D., Keplinger, L., Canfield, S., Gopidi, R., Wakefield, B. J., Koopman, R. J., Belden, J. L., Kruse, R., and Kochendorfer, K. M. (2010), "Issues and questions to consider in implementing secure electronic patient provider web portal communications systems", International Journal of Medical Informatics, 79(7), 469-477.

노 시 춘 (Si Choon Noh)

고려대학교에서 경영정보학(MIS)석사 학위와 경기대학교에서 정보보호 기술공학 박사를 취득했다. KT 초대 시스템보안부장으로 공기업 보안체계를 최초 구축하고 KT 충청전산국장 등 KT IT분야에서 근무했다. 남서울대학교 컴퓨터학과에서 교수로 근무한 이후 현재는 남서울대학교 컴퓨터학과 외래강사와 한국디지털융합직업전문학교 교수로 출강 중이다. 주요 관심분야는 컴퓨터네트워크, 정보보호, 웹기반 의료정보시스템, 인터넷윤리 등이다.

문 송 철 (Songchul Moon)

인하대학교에서 회계학과(경영)학사 학위와 KAIST 경영정보공학과에서 공학석사를 취득했다. 국민대학교 대학원 정보관리학과(정보관리학박사), 한보정보통신(주) 한보철강SI사업부장, 관리이사로 있었으며 가나시스텍(주) 사장, 정보시스템 감리원으로 근무하였다. 현재는 남서울대학교 컴퓨터학과 교수이며, 주요 관심분야는 소프트웨어공학, 시스템 분석 및 설계, 정보시스템 감리 등이다.

송 은 지 (Eunjee Song)

숙명여자대학교 수학과에서 이학사 학위를 취득하였으며, 일본 나고야(名古屋) 국립대학 정보공학과에서 공학석사와 공학박사 학위를 취득하였다. 일본 나고야(名古屋)국립대학 정보공학과 객원 연구원, KIST시스템공학연구소 연구원, 파스퇴르유업 전산실장으로 근무하였으며, 현재는 남서울대학교 컴퓨터학과 교수이다. 주요 관심분야는 IT융합, 빅데이터, 암호학, 수치해석 등이다.

A Study of Field Application Process of Public Key Algorithm RSA Based on Mathematical Principles and Characteristics through a Diagnostic

SiChoon Noh* · EunJee Song** · SongChul Moon***

ABSTRACT

The RSA public key encryption algorithm, a few, key generation, factoring, the Euler function, key setup, a joint expression law, the application process are serial indexes.

The foundation of such algorithms are mathematical principles. The first concept from mathematics principle is applied from how to obtain a minority. It is to obtain a product of two very large prime numbers, but readily tracking station the original two prime number, the product are used in a very hard principles. If a very large prime numbers p and q to obtain, then the product is the two n = p×q easy station, a method for tracking the number of p and q from n synthesis and it is substantially impossible. The RSA encryption algorithm, the number of digits in order to implement the inverse calculation is difficult mathematical one-way function and uses the integer factorization problem of a large amount.

Factoring the concept of the calculation of the mod is difficult to use in addition to the problem in the reverse direction. But the interests of the encryption algorithm implementation usually are focused on introducing the film the first time you use encryption

algorithm but we have to know how to go through some process applied to the field work

This study presents a field force applied encryption process scheme based on public key algorithms attribute diagnosis.

Keywords: RSA Public Key Encryption Algorithm, DES Private Key Algorithm, Mathematical Principles

* Namseoul University, Department of Computer Science, Professor, nsc321@nsu.ac.kr

** Namseoul University, Department of Computer Science, Professor, sej@nsu.ac.kr

*** Corresponding Author, Namseoul University, Department of Computer Science, Professor, moon@nsu.ac.kr