# 효과적인 RSA 암호 알고리즘 교육을 위한 학습 소프트웨어 개발

이동범† · 최명균†† · 곽 진†††


## 요 약

IT기술의 발전으로 인해 언제 어디서나 다양한 정보를 제공 받을 수 있는 정보화 시대가 도래하였지만, 이에 대한 역기능으로 원하지 않는 개인정보 유출 피해가 증가하고 있다. 이러한 개인정보 유출 피해를 방지하기 위해서 사용되는 기본적인 학문이 암호학이다. 하지만 암호학은 복잡한 수학적 이론이 접목되어 있는 학문이기 때문에 많은 사람들이 학습에 많은 어려움을 겪고 있다. 따라서 본 논문에서는 암호학에 대한 이해를 개선하기 위해 일반적으로 전자 서명에서 주로 사용되고 있는 RSA 암호 알고리즘에 대한 지식을 향상 시키는데 도움을 줄 수 있는 소프트웨어를 개발하였다. 개발한 소프트웨어를 통해 RSA 암호 알고리즘의 동작 방식에 대한 이해를 돕고자 한다.


**주제어** : RSA(Riverst Shamir Adleman), 암호, 학습 소프트웨어, 정보보안

# Development of Learning Software for Effective RSA Cryptography Algorithm Education

Dongbum Lee† · Myeonggyun Choi†† · Jin Kwak†††


## ABSTRACT

Recently, by the development of information technology, we can get various information from anywhere in real time. However, personal information is exposed to threats which may incur unwanted information leakage. Cryptography serves as a primary study to prevent this leakage. However, some theories of cryptography are based on complex mathematical theories which make many people confused. Therefore, in this paper, we develope a software which is helpful to understand RSA algorithm, which is widely used algorithm in digital signature to protect personal information.


**Keywords** : RSA(Riverst Shamir Adleman), Cryptography, Learning Software, Information Security

† 준 회 원: 순천향대학교 정보보호학과 박사과정
†† 준 회 원: 순천향대학교 정보보호학과 학사과정
††† 종신회원: 순천향대학교 정보보호학과 교수(교신저자)
논문접수: 2011년 06월 23일, 심사완료: 2011년 06월 29일

# 1. 서 론

IT기술의 발전으로 인해 언제 어디서나 정보를 접할 수 있고 다양한 서비스를 제공 받는 정보화 사회로 빠르게 변화하고 있다. 정보화 사회에서는 시간, 공간의 제약 없이 사용자가 원하는 환경에서 원하는 서비스를 제공 받을 수 있다. 하지만 그 이면에는 도청, 개인정보 유출 등의 보안 위협이 존재하고 있다[1]. 이와 같은 보안 위협에 대응하기 위한 기본적인 분야가 암호학이라고 할 수 있다.

암호학은 현대사회에서 군사, 외교, 개인의 프라이버시 보호, 인터넷 범죄 예방 등에 이용되고 있다. 최근 A은행의 전산 시스템 마비, B캐피탈 개인정보 유출 등 개인 정보 유출 사례가 증가함에 따라 암호학에 대한 관심이 증가하고 있으며 전문 인력뿐만 아니라 일반인에게도 암호학에 대한 지식이 요구되고 있다. 하지만 암호학은 일반인에게 있어서 이해하기 쉽지 않은 분야이다. 따라서 본 논문에서는 암호학에 대한 이해를 개선하기 위해 일반적으로 전자 서명에서 주로 쓰이고 있는 RSA 암호 알고리즘에 대하여 학습 할 수 있는 소프트웨어를 개발하고자 한다. 이를 위해 본 논문에서는 RSA의 수학적 배경에 이용되는 수학 이론을 알기 쉽게 설명하였으며, RSA 알고리즘의 암·복호화가 어떠한 과정을 통하여 동작하는지 그 과정을 분석하고, 스스로 암·복호화를 수행하면서 RSA 알고리즘의 이해를 도울 수 있도록 한다.

본 논문에서는 RSA 암호 알고리즘 학습 소프트웨어를 제안 및 구현하기 위하여 RSA 알고리즘의 수학적 배경부터, RSA 암호 알고리즘의 키 생성과정, 암·복호화 동작 과정을 설명한다. 또한, 학습 소프트웨어를 개발하기 위하여 필요한 C언어, C++, MFC 등의 프로그래밍 언어를 이용하여 구현한다. 이러한 구현과정을 거쳐 구현된 소프트웨어를 통해 RSA 암호 알고리즘의 학습 방법을 제안하고 일반인들에게 암호 알고리즘의 동작 방식을 알기 쉽게 이해시킬 수 있는 RSA 암호 알고리즘 학습 소프트웨어를 개발하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 RSA 암호 알고리즘의 수학적 배경에 대하여 분석한다. 3장에서는 RSA 암호 알고리즘 학습 소프트웨어를 설계하고, 파일 암·복호화, 문장 암·복호화, 전자 서명 및 검증을 프로그래밍 언어로 구현하고, 마지막으로 4장에서는 결론을 맺는다.

# 2. 관련연구

## 2.1 수학적 배경

### 2.1.1 Euler 함수

Euler 함수는 RSA 알고리즘 중 키생성 부분에 사용되는 함수로써 일반적으로 $\phi(n)$으로 표기한다. Euler 함수는 1부터 $n$까지의 양의 정수 중에서 $n$과 서로소인 정수의 개수를 나타내는 함수이며 다음과 같은 성질을 갖는다[3].

- □ $p$가 소수일 때, $\phi(p)=p-1$
- □ $m, n$이 서로 소인 정수일 때, $\phi(mn)=\phi(m)\,\phi(n)$ (곱셈 함수)
- □ 잉여환 $Z/mZ$ 으로 이루어진 군의 위수는 $\phi(m)$이다.
- □ $G$를 위수 $n$인 순환군이라 하자. $n$의 약수 $r$에 대해, 위수가 $r$인 $G$의 원소는 $\phi(r)$개 존재한다.
- □ 순환군 $G$의 생성원은 $\phi(n)$개 존재한다.

### 2.1.2 유클리드 호제법

유클리드 호제법은 2개의 자연수 또는 정식(문자에 대하여 덧셈・뺄셈・곱셈만의 연산을 사용하여 얻어지는 대수식 또는 분모나 근호 속에 문자를 포함하고 있지 않은 식)의 최대공약수를 구하는 알고리즘이다. 호제법이란 두 수가 서로 상대방 수를 나누어서 결국 원하는 수를 얻는 알고리즘을 나타낸다. 2개의 자연수(또는 정식) $a, b$에 대해 $a$를 $b$로 나눈 나머지를 $r$이라 하면, $a$와 $b$의 최대공약수는 $b$와 $r$의 최대공약수와 같다. 이 성질에 따라 $b$를 $r$로 나눈 나머지 $r'$을 구하고, 다시 $r$을 $r'$로 나눈 나머지를 구하는 과정을 반복하여 나머지가 0이 되었을 때 나누는 수가 $a$와 $b$의 최대공약수이다. 과정을 수식으로 나타내면 다

음과 같다.

두 양의 정수 $a, b$에 대하여

$$
\begin{aligned}
b &= aq_1 + r_1 && 0 < r_1 < a \\
a &= r_1 q_2 + r_2 && 0 < r_2 < r_1 \\
r_1 &= r_2 q_3 + r_3 && 0 < r_3 < r_2 \\
&\vdots \quad \vdots \\
r_{n-3} &= r_{n-2} q_{n-1} + r_{n-1} && 0 < r_{n-1} < r_{n-2} \\
r_{n-2} &= r_{n-1} q + r_n && 0 < r_n \ < r_{n-1} \\
r_{n-1} &= r_n q_{n+1}
\end{aligned}
$$

일 때, $\gcd(a,b) = r_n$이 성립한다[4]. 앞의 등식들을 반대로 이용하면 $\gcd(a,b) = au = bv$인 정수 $u, v$를 확장 유클리드 호제법으로 구할 수 있으며, 구하는 과정은 다음과 같다[5].

$$
\begin{aligned}
r_n &= r_{n-2} - r_{n-1} q_n \\
&= r_{n-2} - (r_{n-3} - r_{n-2} q_{n-1}) q_n \\
&= r_{n-2}(1 + q_{n-1} q_n) + r_{n-3}(-q_n) \\
&\quad \vdots \\
&= au + bv
\end{aligned}
$$

확장 유클리드 호제법은 이러한 연산 과정을 거쳐 RSA의 개인키를 구할 때 사용된다.

### 2.1.3 멱승함수

RSA 알고리즘의 암·복호화 과정에서는 멱승 연산이 사용된다. 이러한 멱승 연산 $X^E \bmod N$을 계산하는 알고리즘으로는 이진 방식, m-ary 방식[6], window 방식[7] 등이 있다. 멱승 연산을 수행하는 가장 일반적인 것으로는 square and multiply 방식으로 알려진 이진 방식이 있다. 이 방식은 $k$비트인 지수 $E$를 이진수로 표현한 후 중간값을 $C$라 두고 이 값을 반복하여 제곱하면서 해당 비트가 1일 때만 $X$를 곱해 준다.

이진 방식에는 지수 비트를 검색하는 방향에 따라 Left-to-Right 방식과 Right-to-Left 방식으로 구분된다. Left-to-Right 방식은 지수를 왼쪽부터 최상위 비트에서 하위로 검색하는 방식이며 Right-to-Left 방식은 반대로 최하위 비트부터 상위 비트로 검색하는 방식이다.

본 논문에서 구현하는 RSA 알고리즘에서는 L-to-R Binary 연산을 수행한다. $X^{13} \bmod N$ 을 L-to-R Binary 연산으로 예를 들어 살펴보면 다음과 같다.

먼저 13을 이진수로 표현하면 1101이다.

```
L-to-R Binary(X, E, C)
1. C=1;
2. for i=0 to k-1 step +1
3.        C=C·C mod N
4.           if E_i≡1 then C=C·T mod N;
5. return (C);
```

□ 알고리즘 과정에서 L-to-R 이진 멱승 방법

ex) $X^{1101}$

$i = k-1$일 때 $C = 1, \quad C = X$
$i = k-2$일 때 $C = X^2, \ C = X^3$
$i = k-3$일 때 $C = X^6$
$i = k-4$일 때 $C = X^{12}, C = X^{13}$
$Return(C = X^{13})$

이진 방식은 지수 $E$가 $k$비트인 경우 $k$번의 제곱과 $HW(E)$번의 곱셈이 필요하다. 여기서 $HW()$는 $E$의 Hamming Weight로써 $E$를 이진수로 나타낼 $HW(E)$때의 1의 개수를 의미한다. 일반적으로 한 비트가 1이 될 확률은 1/2이므로 이진 방법의 곱셈 수는 평균 $1.5k$번이 되고 최대 $2k$번이 된다[8]. 예를 들면 1024비트의 지수에 대해 평균적으로 1,536번의 모듈러 곱셈이 필요하게 된다. 이 방식은 구현하기 용이할 뿐만 아니라 결과값을 저장하기 위한 메모리 이외의 소요 메모리가 필요가 없어 효과적이지만 다른 방식에 비해 속도가 늦다는 단점이 있다.

## 2.2 RSA 알고리즘

RSA 알고리즘은 1978년 Riverst, Shamir, Adleman 이라는 3명의 MIT대학 교수에 의해 제안된 방식이다[9]. RSA 암호 알고리즘은 큰 소수로 된 합성수의 소인수분해를 하는 것에 기반을 둔 암호 알고리즘이다. RSA 암호 알고리즘은 1978년에 고안된 블록암호로써 키의 분배와 관리

의 용이, 인증과 전자 서명이 가능하다는 장점으로 인해 인정받은 알고리즘이다.

### 2.2.1 키생성 알고리즘

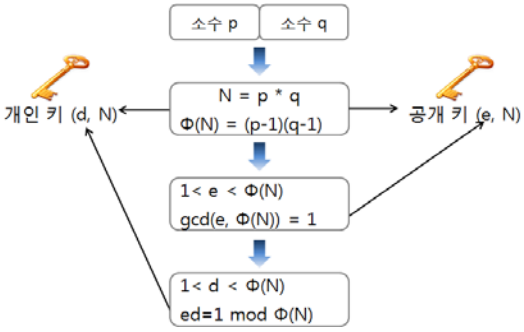
RSA 알고리즘의 키는 공개키와 개인키로 구분되어 있다. 공개키는 공개목록에 등록하여 공개하고 개인키는 개인이 비밀리에 보관한다. 키는 공개키부터 생성되며 RSA의 키 생성과정은 다음과 같다[10].

- ☐ 큰 소수 $p, q$를 선택
- ☐ $n = pq$, $\phi(n) = (p-1)(q-1)$
- ☐ $1 < e < \phi(n)$, $\gcd(e, \phi(n)) = 1$을 만족한 정수 $e$를 선택
- ☐ $1 < e < \phi(n)$, $ed = 1 (\mathrm{mod}\, \phi(n))$을 만족하는 유일한 정수 $d$를 구함
- ☐ 공개키 : $(n, e)$, 개인키 : $(d)$

공개키를 생성하기 위해서 먼저 소수 $p, q$를 선택한다. 안전성을 위해 512비트 이상의 수와 두 소수 $p, q$의 길이를 비슷하게 선택한 다음 두 수의 곱인 합성수 $N$을 구하고 $\phi(N)$을 구한다. $\phi(N)$은 앞에서 설명한 수학적 배경의 Euler 함수를 사용하여 구하며 $(p-1)(q-1)$=$\phi(N)$을 통해 구한다. $N$과 $\phi(N)$을 구한 후에 $\gcd(e, \phi(N)) = 1$을 만족하는 $e$를 공개키로 선택한다.

개인키를 구하는 과정은 선택한 $e$에서부터 확장 유클리드 알고리즘을 통해 $ed \equiv 1 \bmod \phi(N)$을 만족하는 $d$를 구해서 개인키로 선택한다.

RSA의 키 생성알고리즘에 대한 전체적인 흐름도는 <그림 2-1>과 같다.



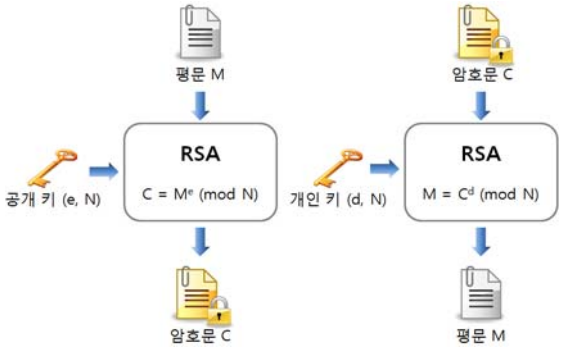

<그림 2-1> RSA 키생성 알고리즘

### 2.2.2 암·복호화 알고리즘

RSA 알고리즘의 암·복호화 연산은 멱승연산에 의해 수행된다. 평문은 블록으로 암호화되며 각 블록은 $n$(키 값의 곱)보다 작은 Binary 값을 가진다. RSA 알고리즘의 암·복호화 과정은 평문 블록$M$과 암호문 블록$C$에 대하여 다음의 형태를 따른다.

$$C = M^e \bmod n \text{ (암호화 과정)}$$
$$M = C^d \bmod n \text{ (복호화 과정)}$$

<그림 2-2>는 RSA 알고리즘의 암·복호화 과정을 그림으로 나타내었다. 사용자 $A$가 메시지 $M$을 사용자 $B$에게 전송하고자 할 때 $A$는 $C = M^e \bmod n$을 계산하여 얻은 암호문 $C$를 전송한다. 사용자 $B$는 전송 받은 암호문 $C$에 대하여 $M = C^d \bmod n$을 계산하여 복호문 $M$을 얻는다[11].



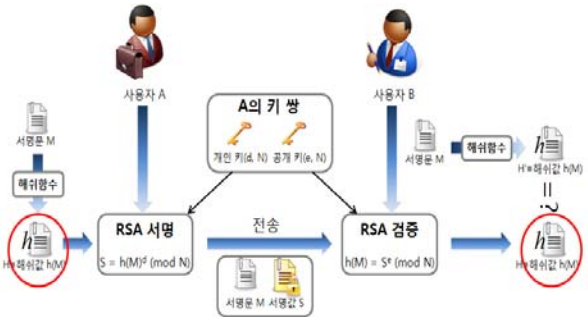

<그림 2-2> RSA 알고리즘 암·복호화

### 2.2.3 전자 서명

RSA 알고리즘의 공개키($e$, N)와 개인키($d$, N)가 mod $n$의 곱셈에 대한 역원 관계에 있으므로 공개키로 암호화 한 메시지는 개인키로 복호화할 수 있으며 개인키로 암호화한 메시지는 공개키로 복호화할 수 있다. 결과적으로 상대방의 공개키를 이용하여 암호화를 하기 때문에 특정 상대방만이 복호화를 하여 평문을 얻을 수 있다. 이를 응용하여 자신의 개인키를 이용한 암호화는 자신임을 입증하는 전자 서명에 사용되고 있다[2].

서명자는 문서의 데이터를 해쉬함수를 통해 해쉬값을 생성하고, 생성된 해쉬값을 개인키로 암호

화 한 후 문서에 서명한다. 검증절차에서는 서명 부분만 획득하여 공개키로 복호화하고 문서에 서명을 제외한 데이터의 해쉬값을 저장한다. 이 두 가지 값을 비교하여 동일하면 검증이 된 문서이고, 그렇지 않으면 검증이 되지 않은 문서이다 [12]. 이러한 과정을 그림으로 살펴보면 (그림 2-3)과 같다.



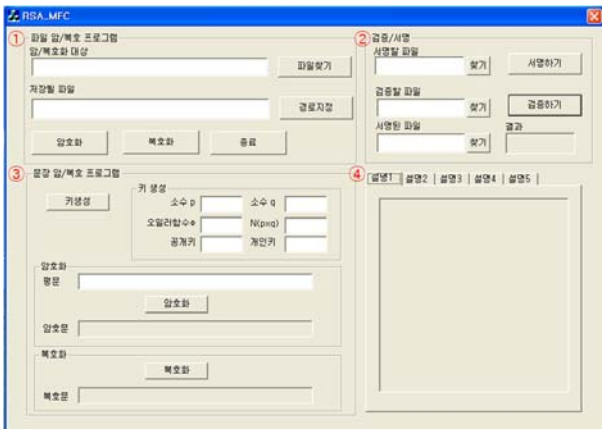

<그림 2-3> RSA 서명생성 및 검증

## 3. 소프트웨어 설계 및 구현

### 3.1 MFC 인터페이스 설계

구현하는 학습 소프트웨어는 RSA 암호 알고리즘의 동작 방식을 학습자가 쉽게 이해할 수 있도록 한다. 이론 위주의 학습방식을 벗어나 실제로 학습자가 원하는 문장을 암호화하고, 이 문장을 이용하여 암·복호화 되는 과정을 보여준다. RSA 암호 알고리즘을 학습할 때 어려움을 많이 겪는 수학적 배경을 중점적으로 구현함으로써 학습하려는 학습자가 암·복호화 과정을 단계별로 학습할 수 있어 효율적인 학습을 할 수 있도록 한다. 또한 파일에 있는 내용을 직접 암·복호화할 수 있으며, 어떤 식으로 전자 서명이 이루어지는지 파일로 서명값을 넣어서 검증할 수 있도록 하였다. 구현하고자 하는 RSA 알고리즘 학습 소프트웨어는 <그림 3-1>과 같으며 ①~④의 기능은 다음과 같다.

① 파일의 내용을 암·복호화
② 전자 서명에서의 서명 및 검증을 수행
③, ④ 실질적인 학습소프트웨어의 주 기능으로 문장을 암·복호화 및 그에 대한 설명부분으로써 키생성, 암·복호 과정을 설명한다.



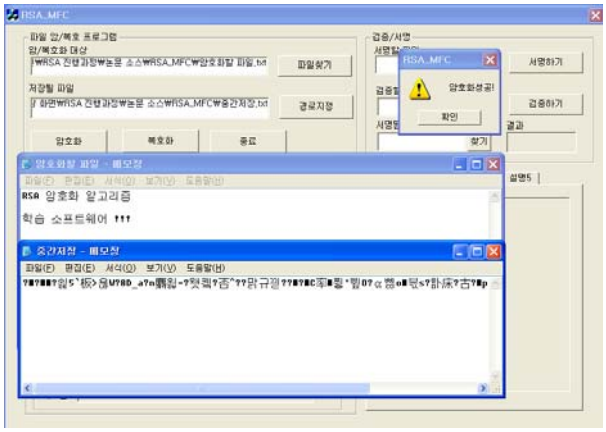

<그림 3-1> RSA 알고리즘 학습 소프트웨어

### 3.2 RSA 학습 소프트웨어 구현

#### 3.2.1 파일 암·복호화

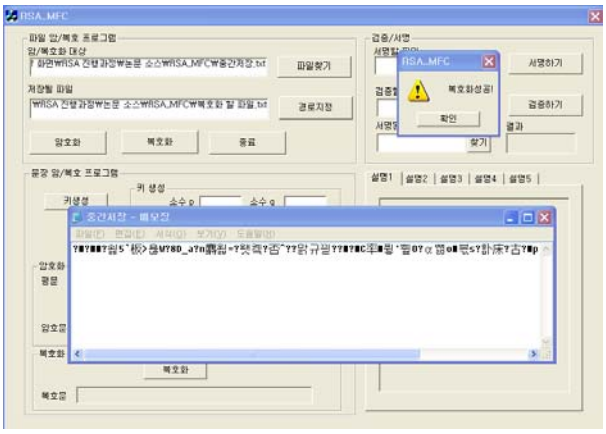
학습자가 암호화 할 파일을 선택하고 저장할 파일명이나 경로를 지정한 후에 암호화 버튼을 누르면 <그림 3-2>와 같이 암호화가 성공하였다는 메시지가 출력되면서 암호화된 파일이 생성된다.

그 후 암호화된 파일을 확인해 보면 <그림 3-2>와 같이 암호화된 형태로 출력되어 원본 파일의 내용을 확인할 수가 없다.



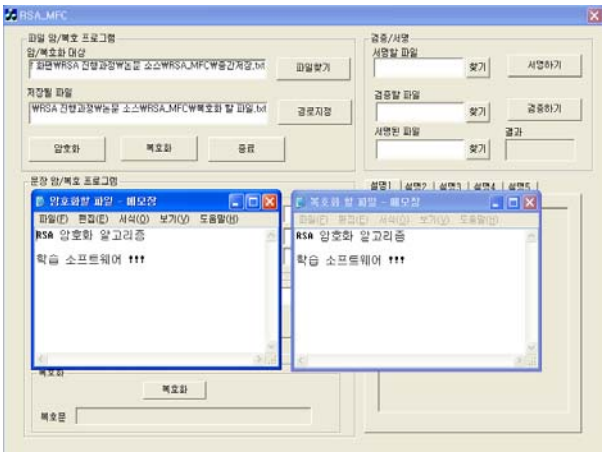

<그림 3-2> 암호화 수행 화면

복호화 과정은 암호화 과정과 동일하게 복호화 할 파일을 선택한 후에 저장할 파일이나 경로를 지정한 다음 복호화 버튼을 누르면 <그림 3-3>과 같이 복호화가 성공하였다는 메시지가 출력되면서 파일이 복호화 된다.



<그림 3-3> 복호화 수행 화면

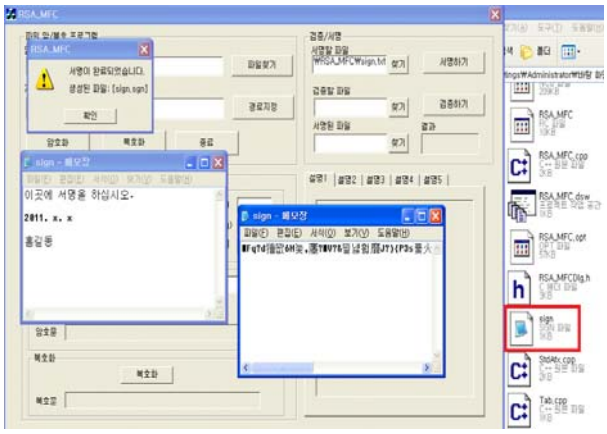
복호화 과정을 수행한 후에 정상적으로 암·복호화가 이루어졌는지 확인하기 위해 <그림 3-4>와 같이 비교한다. 암호화하기 전 파일과 복호화를 거친 파일을 비교해 봤을때 동일하다면 정상적으로 암·복호화 과정이 이루어진 것이며 만약, 복호화를 거친 파일의 내용을 확인할 수 없다면 복호화 과정이 정상적으로 이루어지지 않은 것이다.



<그림 3-4> 파일 비교

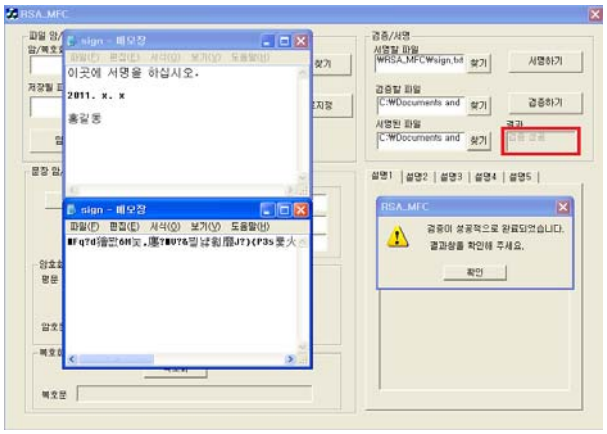
### 3.2.2 파일 서명·검증

<그림 3-5>는 RSA 암호 알고리즘 학습 소프트웨어를 이용한 파일의 서명·검증 과정이다. 파일의 서명·검증 부분은 파일 암·복호화 부분과 동일하게 학습자가 서명할 파일을 지정하고 서명하기 버튼을 누르면 다음과 같이 서명이 완료되었다는 메시지가 출력 되며 확장자가 '.sgn'인 서명된 파일이 생성된다.



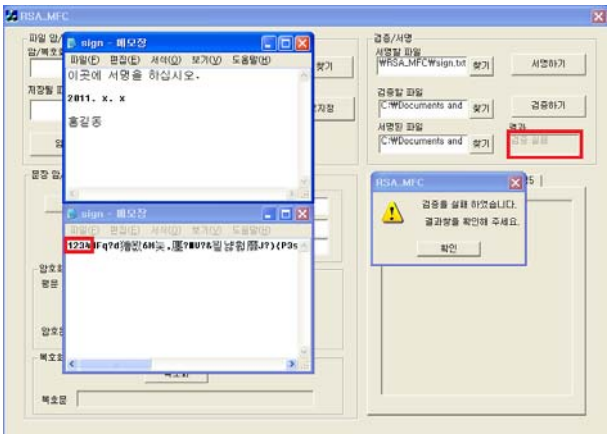
<그림 3-5> 서명 과정

서명이 완료된 것을 확인하였으면 이제 검증 과정을 거치게 되는데 검증 과정은 검증할 파일인 원래의 서명할 파일과 동일한 파일을 선택한다. 서명된 파일에는 서명 과정에서 생성된 확장자 '.sgn' 파일을 선택한다. 선택한 다음 검증하기 버튼을 누르면 검증이 성공적으로 완료되었다는 메시지가 출력되면서 결과 창에 결과 값이 출력된다. 파일 검증 절차는 <그림 3-6>과 같다.

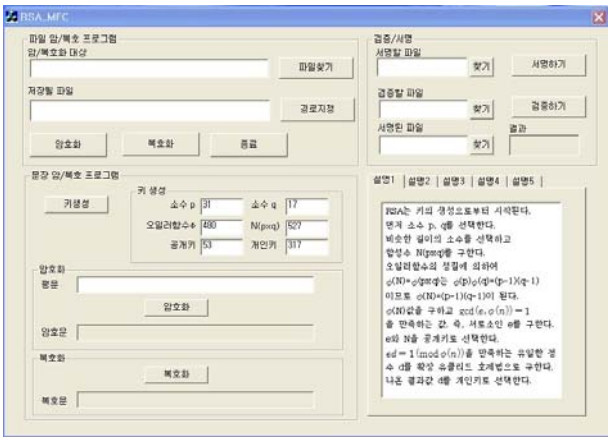


<그림 3-6> 검증 과정

만약 생성된 확장자 '.sgn'파일이 도중에 변경되었거나 손상 되었을 경우에는 검증이 실패된다. <그림 3-7>에서는 서명된 파일이 변경 되었거나 손상 되었을 경우에도 정확하게 검증을 할 수 있는지 확인하기 위하여 임의로 서명된 파일 앞부분에 '1234'라는 숫자를 삽입하여 검증 과정을 진행하였다.



<그림 3-7> 검증 실패



<그림 3-8> 키생성 설명 I

<그림 3-7>과 같이 서명된 파일이 조금만 변경되더라도 검증이 실패했다는 메시지와 함께 서명을 검증 하지 못한다.

구현하는 RSA 알고리즘 학습 소프트웨어에서는 실생활에서도 응용되는 RSA의 전자 서명방식을 학습자가 직접 서명하고 검증함으로써 전자서명이 어떠한 식으로 서명되고 검증하는지 쉽게 이해 할 수 있도록 하였다.

### 3.2.3 문장 암·복호화

문장 암·복호화 부분은 RSA 암호 알고리즘 학습 소프트웨어의 핵심 부분이다. 먼저 학습자가 키생성 버튼을 누르면 키생성 알고리즘을 거쳐 키가 랜덤으로 생성이 된다. 키생성 버튼을 눌렀을 때 설명1이 나오고 설명2까지 학습자가 RSA의 키생성 부분에 대한 이해를 돕기 위해서 알기 쉽게 설명한 내용들을 볼 수 있다. 설명1의 내용에는 키가 생성되는 과정을 설명 하였으며, 설명2에서는 설명1에서 설명한 알고리즘을 간단한 숫자로 예를 들어서 나타내었다.

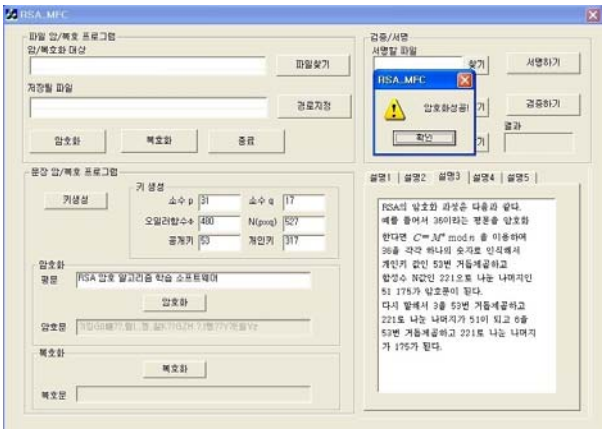
키생성 부분에 대한 그림은 <그림 3-8>과 <그림 3-9>와 같다.

<그림 3-9> 키생성 설명 II

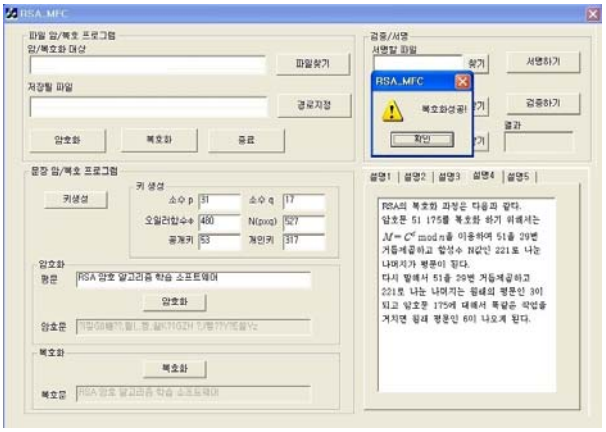
<그림 3-10>은 문장에 대한 암호화 과정이다. 암호화 버튼을 눌렀을 때 설명3이 출력되면서 암호화가 수행되는 과정을 간단한 숫자로 예를 들어서 설명한 것을 볼 수 있다. 위와 같은 설명을 참고하여 직접 문장을 암호화함으로써 암호화 과정을 쉽게 이해할 수 있다. 암호화 과정이 성공적으로 수행되었다면 원본 문장을 알아 볼 수 없도록 출력되고, 암호문을 수정하지 못하게 하기 위해 MFC 인터페이스 Dialog 상에서 암호문 Edit Box에 Disabled 설정을 추가하였다.

<그림 3-11>은 문장에 대한 복호화 과정이다. 복호화 버튼을 눌렀을 때 설명4가 나오면서 복호화가 수행되는 과정을 간단한 숫자로 예를 들어서 설명한 것을 볼 수 있다. 정상적으로 복호화 과정이 수행되었는지를 판별하기 위해서는 복호화된 결과와 처음 입력한 평문의 내용이 일치하

는지 확인하면 된다.



<그림 3-10> 문장 암호화 과정



<그림 3-11> 문장 복호화 과정

## 4. 결 론

개인정보 유출 피해가 증가함에 따라 전자 서명, 인증 분야 등에 많이 활용되고 있는 암호학 분야에 대한 관심이 증가하고 있다. 따라서 본 논문에서는 공개키 암호 방식 중에서 많이 사용되고 있는 RSA 암호 알고리즘에 대한 학습자의 쉬운 이해를 돕기 위해서 학습 소프트웨어를 개발하였다.

본 논문에서는 RSA 암호 알고리즘을 분석하여 설계 및 구현하였으며, 학습자들이 이해하기 쉽도록 예를 들어 설명하였다.

학습자의 기존 RSA 암호 알고리즘 학습방식에는 NIST 표준 문서, RSA 관련 논문, 암호학 관련 서적 등을 참고하는 방식이 있다. 하지만 논문이나 참고하는 서적에는 이해하기 어려운 수식들로만 설명이 되어있어 학습자가 RSA 암호 알고리즘을 학습하는데 있어서 많은 어려움을 겪고 있다. 하지만 본 논문에서 구현한 학습 소프트웨어는 암·복호화 과정에 따라 부가적으로 예를 들어 설명하였다. 이를 바탕으로 전공분야 학습자의 RSA에 대한 쉬운 이해와 흥미로운 학습을 진행할 수 있게 하며, 일반 학습자들에게는 암호 알고리즘 및 암호학에 대한 거부감을 해소할 수 있을 것으로 생각된다. 또한 암·복호화 및 전자 서명 과정을 학습자들이 직접 실행함으로써 RSA 암호 알고리즘의 동작과정과 RSA가 주로 응용되는 전자 서명과정을 쉽게 이해할 수 있으며, 학습자들의 암호학에 대한 관심을 높일 수 있을 것으로 기대된다.

## 참 고 문 헌


[1] 김성식, 조성환(2008). 학습자 입장에서 살펴본 교원양성 대학에서의 정보윤리교육을 위한 교육과정 모델 설계. **컴퓨터교육학회 논문지, 11**(3), 34-35.

[2] 조광문(2004). 전자상거래에서 안전한 정보교환을 위한 웹 서비스 기반의 XML 보안 모델. **컴퓨터교육학회 논문지, 7**(5), 93-95.

[3] William D. Banks, John B. Friedlander, Carl Pomerance &Igor E. Shparlinski(2004). *Multiplicative structure of values of the euler function.* Fields Institute Communications vol. 41. 29-31.

[4] B. Vall´ee,(1998). *The complete analysis of the binary euclidean algorithm.* Lecture Notes in Computer Science 1423. 77-94.

[5] S. P. Glasby(1999). *Extended Euclid''s algorithm via backward recurrence relations.* Mathematics Magazine. 72. 228-230.

[6] Okeya K, &Sakurai K(2002). *A Second-Order DPA Attack Breaks aWindow-method based Countermeasure against Side Channel Attacks.* In Proceedings of Information Security

Conference-ISC. 389-401.
[7] MahnKi Ahn, JaeCheol Ha, HoonJae Lee, &SangJae Moon(2003). *A Random M-ary Method based Countermeasure against Side Channel Attacks.* Information Security Institute, 3(3). 36-37.
[8] Sung-Ming Yen, Seungjoo Kim, Seongan Lim, &Sangjae Moon(2002). *A Countermeasure against One Physical Cryptanalysis May Benefit Another Attack.* Computer Science, Volume 2288/2002. 417-418.
[9] R. L. Rivest, A. Shamir, &L. Adleman (1978). *A Method for Obtaining Digital Signautres and Public Key Cryptosystems.* Comm. of ACM vol. 21. 8-10.
[10] Thierry Congos, &Francois Lesueur(2009). *Experimenting with Distributed Generation of RSA Keys.* Elsevier Science B. V. 1-3.
[11] Latif K, Mahboob A, &Ikram N(2009). *A parameterized design of Modular Exponentiation on reconfigurable platforms for RSA cryptographic processor.* 2nd International Conference on. 1-3.
[12] NIST(2009). *Digital Signature Standard.* FIPS PUB 186-3. 22-25.

**이 동 범**

2008 순천향대학교 정보보호학과 (정보보호학 학사)
2010 순천향대학교 정보보호학과 (정보보호학 석사)
2010～현재 순천향대학교 정보보호학과 박사과정
관심분야: 컴퓨터교육, 정보보호, 보안성 평가
E-Mail: dblee@sch.ac.kr

**최 명 균**

2006～현재 순천향대학교 정보보호학과(학사과정)
관심분야: 컴퓨터교육, 정보보호, 암호 프로토콜 등
E-Mail: mgchoi@sch.ac.kr

**곽 진**

1994～2006 성균관대학교(공학사, 공학석사, 공학박사)
2006～2006 일본 큐슈대학교 방문연구원
2006～2006 일본 큐슈시스템 정보기술연구소 특별연구원
2006～2007 정보통신부 개인정보보호기획단 개인정보보호팀 통신사무관
2007～2009 정보통신연구진흥원 집필위원
2007～현재 순천향대학교 정보보호학과 교수
2009～2009 순천향대학교 공과대학 교학부장
2009～2010 순천향대학교 정보보호학과 학과장
2010～2010 교육과학기술부 국가기술수준평가 전문위원
현재: 정보통신산업진흥원 기술평가위원, 사)국제정보능력평가원 쇼핑몰 플래너 자격 검정 출제 및 채점위원, 한국과학기술정보연구원 충남 과학기술 정보협의회 전문위원, 지식경제부 지식경제기술혁신평가단 평가위원, 순천향BIT 창업보육센터 센터장, 순천향대학교 중소기업산학협력센터 센터장
관심분야: 암호프로토콜, 응용시스템보안, 개인정보보호, 정보보호제품평가, 클라우드 컴퓨팅보안 등
E-Mail: jkwak@sch.ac.kr