<RSA 구현 과제>

12161595 배재경

30 bit의 RSA 암호 알고리즘과 RSA 기반의 Digital Signature을 구현하는 프로그램을 만들었다. 구현 환경은 Visual Studio 2017, 구현 언어는 C++이다.

RSA의 공개키를 암호화하기 위해 우선 키 생성 알고리즘에서 10bit의 소수인 p, q, r을 랜덤하게 생성한다. 랜덤한 p, q, r 값이 소수인지 확인하기 위해서 Miller-Rabin 소수판정법을 MillerRabin 함수로 구현하였다. MillerRabin에 의해 false가 리턴된 값은 다시 랜덤한 값을 구하도록 하였다.

확인할 값을 n이라 하고, n-1을 m이라 하면 m은 2의 k제곱으로 분해할 수 있고, 남은 값은 그대로 사용한다. a를 1보다 크고 n-1보다 작은 랜덤한 값으로 정하고, a와 n의 최대공약수가 1이 아니면 소수가 아니므로 false를 리턴한다. b는 a에 남은 m값을 제곱하여 mod n한 값이다. 제곱하여 모듈러한 값은 expo 함수로 구현하였다. b가 1과 n-1이 아니면 b*b mod n이 되게 하고, b가 n-1일 경우 반복문을 빠져나오게 한다. b가 n-1이 아닐 경우 소수가 아니므로 false를 리턴한다. 이를 20번 반복한 뒤에는 소수일 가능성이 매우 높아지므로 소수라고 판정하여 true를 리턴한다.

N은 p*q*r, Euler's totient(phi)는 (p-1)*(q-1)*(r-1)로 한다. 공개키 e는 1보다 크고 phi보다 작은 수 중에서 mod phi에 대한 역원을 갖는 수, 즉 최대공약수가 1인 서로소를 랜덤하게 뽑았다. 그리고 공개키의 역원을 비밀키 d로 한다. 역원은 확장된 유클리드 호제법을 구현한 extended 함수를 사용하여 구할 수 있게 하였다.

메시지를 입력하면 입력된 메시지 message를 보여주고, message 값에 공개키를 제곱하고 mod N하여 암호화 한다. 암호화 된 값을 복호화 하는 데 걸리는 시간을 단축하기 위해서 중국인의 나머지 정리를 구현한 crt 함수를 사용한다. 세 소수를 사용하는 CRT는 두 소수 p, q를 combine한 뒤 r을 combine하여 구할 수 있다. d mod r-1을 dr로, 정하고 암호화 한 값에 dr을 제곱하고 mod r한 값을 mr로 정하고, mod p*q인 r의 역원을 구한다. 같은 방식으로 dpq와 mpq를 구하고 combine하면 복호화 된 메시지의 값인 dec_cipher를 구할 수 있다.

복호화 된 메시지 값이 변조되었는지 확인하기 위해서 전자 서명을 사용한다. 해시 함수로 message 값의 해시 값을 hash로 하였다. 전송자가 hash에 d를 제곱하고 mod N하여 암호화된 signature 값을 수신자에게 보내면, 수신자가 dec_cipher의 해시 값 dhash를 구하고, signature에 e를 제곱하고 mod N하여 복호화 된 signature 값 dsignature를 구한다. 수신자는 dhash와 dsignature을 비교하고 서로 같으면 복호화 된 메시지 값이 유효하다는 것이고, 같지 않으면 유효하지 않다는 것을 확인할 수 있다. 이로서 서명이 검증되고 나면 프로그램이 종료된다.