제 18 회 한국정보처리학회 추계학술발표대회 논문집 제 9 권 제 2 호 (2002.11)

안전성이 증명 가능한 공개키 암호 시스템을 이용한 E-mail 어플리케이션 구현

안기범*, 이승우*, 오수현*, 원동호*
*성균관대학교 정보통신공학부
e-mail : {gbahn, swlee, shoh, dhwon}@dosan.skku.ac.kr

The E-mail Application Implementation Using Provably Secure Public Key Cryptosystem

Ki-Bum Ahn*, Seung-Woo Lee*, Soo-Hyun Oh*, Dongho Won*
*School of Information & Communication Engineering, Sungkyunkwan University

요 약

최근 암호문을 변경할 수 없다는 안전성(CCS)을 제공하는 공개키 암호 알고리즘에 대한 필요성이 제기되었고, 그에 따른 안전성이 증명 가능한 공개키 암호 방식을 개발하려는 연구가 활발히 진행되고있다. 본 논문에서는 [7]에서 제안한 안전성이 증명 가능한 공개키 암호시스템을 이용하여 안전한 E-mail 어플리케이션을 구현하였다. 구현한 E-mail 어플리케이션은 사용자가 자신이 원하는 공개키 알고리즘, 대칭키 알고리즘, 해쉬 알고리즘 등을 선택하여 사용할 수 있고, 누구나 쉽게 사용할 수 있도록 인터페이스를 구성 하였으며 E-mail 과 접목하여 활용 가능하다는 장점이 있다.

1. 서론

1976 년Diffie와 Hellman이 공개키 암호 시스템을 처음으로 제안한 이후 많은 공개키 암호 시스템이 개발되고 이에 공개키 암호 시스템의 안정성을 증명 하려는 연구가 계속 되었다[1]. 1984 년 Goldwasser와 Micali는 수동적인 공격자를 모델로 한 구분 불가능성(Indistinguishability:IND)이라는 증명 가능한 공개키 암호 시스템의 안전성 개념을 제시하였으며[2], 1991 년 Dolev는 능동적 공격자를 모델로 한 조작 불가능성(Non-Malleability:NM)이라는 안전성 개념을 제시하였다[3].

이러한 안전성 개념은 Bellare등에 의해 기존의 선택 평문 공격자(Chosen Plaintext Attack:CPA)와 선택 암호문 공격자(Chosen Ciphertext Attack:CCA1), 적응-선택 암호문 공격자(Adaptive Chosen Ciphertext Attack:CCA2) 모델을 체계화하였고, IND-CCA2 와 NM-CCA2 가 동치임을 증명하였으며 최근 이 두 안전성 개념을 선택 암호문 공격에 안전한 안전성(Chosen Ciphertext Secrity:CCS)으로 부르고 있다[4].*

* 본 연구는 한국과학재단의 목적기초연구(97-0100-13-01-5) 지원사업으로 수행되었음.

본 논문에서는 기존의 공개키 암호 시스템과 대칭키 암호 시스템을 결합하여 CCS를 만족하며 효율적인 공개키 암호 시스템 즉, 표준모델에서 안전성이 증명 가능한 공개키 암호 시스템을 이용하여 실생활에서 많이 사용되고 있는 메일과 쉽게 사용할 수 있도록 메일 어플리케이션을 구현하였다. 본 논문의 구성은 다음과 같다. 먼저 제 2 장에서는 표준모델에서 안전성이 증명 가능한 암호 시스템에 대해 설명하고, 제 3 장에서는 구현한 메일 어플리케이션 암호화/복호화 모듈, 키생성 모듈, 시스템 파라메터 모듈에 대해 설명하며, 4 장에서는 결론을 맺는다.

2. 표준모델에서 안전성이 증명 가능한 공개키 암호 시스템

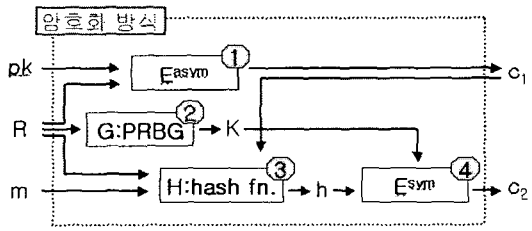
본 논문에서 이용한 공개키 암호 시스템은 기존의 공개키 암호 방식과 대칭키 암호 방식을 결합한 방식으로 다음과 같은 세 개의 알고리즘(K^{hy} , E_{pk}^{hy} , D_{sk}^{hy})으로 구성되며 G 는 의사 난수 생성기, H 는 일반적인 해쉬함수이다.

2.1 키 생성 알고리즘

- $K^{hy}(1^k)$: 키 생성 알고리즘
- 안전성 계수(security parameter) k 을 입력으로 공개키와 비밀키 쌍 (pk, sk) 을 출력한다.

2.2 암호 알고리즘

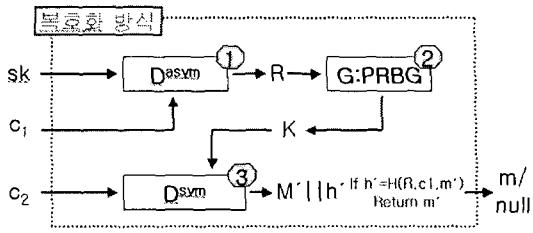
- $E_{pk}^{hy}(m;R,r)$: 암호 알고리즘
- $c_1 = E_{pk}^{asym}(R;r)$: 공개키 암호 방식으로 랜덤한 입력 값 R 을 암호화한다. (r : random coins)
- $K = G(R)$: R 을 입력으로 대칭키 암호 방식의 세션키(Session key) K 을 출력한다.
- $H(R,c_1,m)$: 검증정보로 사용되며, R 과 c_1, m 이 해쉬함수의 입력 값이 된다.
- $c_2 = E_{pk}^{sym}(m \| h)$: 평문 m 과 해쉬함수의 결과 값 h 을 K 을 이용해 대칭키 암호 방식으로 암호화한다.
(* 결정론적 공개키 암호 방식일 경우, 공개키 암호 방식의 입력 값 r 을 null값을 갖는다.)



[그림 1] 암호화 과정

2.3 복호 알고리즘

- $D_{sk}^{hy}(C)$: 암호 알고리즘 $(C = (c_1, c_2))$
- $R = D_{sk}^{asym}(c_1)$: c_1 을 복호해서 R 값을 얻는다.
- $K = G(R)$: 복호된 R 을 이용해 세션키 K 를 출력한다.
- $D_{sk}^{sym}(c_2)$: K 를 이용해 c_2 를 복호한다.
- $D_{sk}^{sym}(c_2) = m' \| h'$ 일 때, $h' = H(R,c_1,m')$ 를 만족하면 $m'(=m)$ 을 출력한다.



[그림 2] 복호화 과정

3. 메일 어플리케이션 구현

본 3 장에서는 구현한 메일 어플리케이션을 각 모듈(암호화, 복호화, 키생성, 시스템 파라메터 생성)별로 기술하며 개발환경은 다음과 같다.

- 운영체제 : Window 95/98/NT
- 개발 언어 : Visual C++ 6.0

3.1 암호화 모듈

암호화 모듈에서는 [그림 3]과 같이 암호화 과정에서 사용하는 알고리즘은 다음과 같다.

- 공개키 알고리즘

- RSA : 합성수의 소인수 분해의 어려움에 기반하는 공개키 암호 알고리즘
- Diffie-Hellman : 이산대수 문제의 어려움에 기반하여 공개된 채널상에서 비밀키 교환에 관한 공개키 알고리즘
- Okamoto-Uchiyama : 특정 부분군(Subgroup)에서 이산대수 문제의 어려움에 기반하는 공개키 알고리즘

- 대칭키 알고리즘

- AES : DES(Data Encryption Standard)를 대체하기 위해 공모에 의해 선정된 대칭키 알고리즘으로 키(Key)는 128, 192, 256 비트의 키를 사용한다.
- SEED : 한국정보보호진흥원에서 우리나라의 표준으로 개발한 대칭키 알고리즘으로 키는 128 비트의 키를 사용한다.

- 해쉬 알고리즘

- HAS160 : 한국형 디지털 서명 표준인 KCDSA(Korea Certification-based Digital Signature)에서 사용될 목적으로 개발되어 메시지(Message)를 512 키트 블록단위로 처리하여 160 비트의 해쉬 코드(Hash Code)로 출력한다.

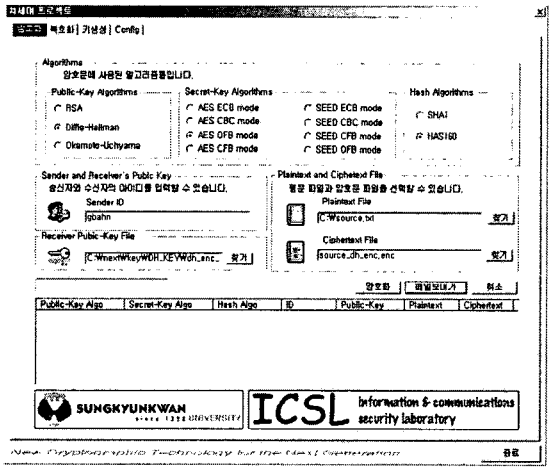
- SHA1 : MD4 를 기반으로 설계되었으며, SHA의 발표되지 않은 결점들이 수정된 개정판이다, 2^{64} 비

트 이하의 메시지에서 160 비트의 해쉬코드를 출력한다.

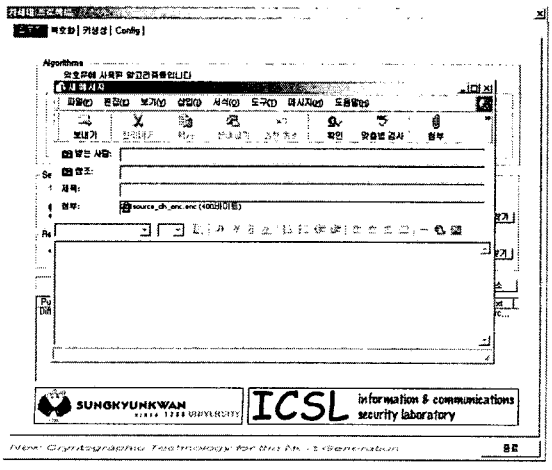
대칭키 알고리즘인, AES(Advanced Encryption Standard), SEED는 각각 네 가지의 운용모드를 적용하며 다음과 같다.

- 대칭키 알고리즘의 운용모드
- ECB 모드(Electronic codebook mode)
- CBC 모드(Cipher block chaining mode)
- CFB 모드 (cipher feedback mode)
- OFB 모드(Output feedback mode)

암호화 과정에서 자신이 암호화 하려는 평문을 수신자의 공개키를 이용하여 암호화 한다. 또한 송신자는 자신의 ID를 암호문에 끝에 삽입하므로서 수신자가 어느 사람으로부터 보내온 암호문인지 알 수 있다.



[그림 3] 암호화 과정



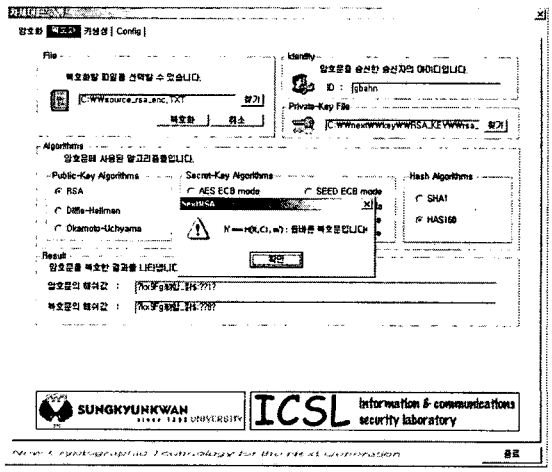
[그림 4] 암호문 메일 보내기

E-mail과의 연동을 고려하여 평문을 암호화 하는 과정을 성공적으로 수행하게 되면 리스트 컨트롤에 나타난 암호문 파일을 선택하여 '메일보내기' 버튼을 이용하여 [그림 4]와 같이 메일을 보낼 수 있다.

3.2 복호화 모듈

복호화 모듈에서는 [그림 5]와 같이 자신이 복호화 하고자 하는 암호문을 입력하고 자신의 비밀키를 입력하므로서 암호문을 복호화 한다. 암호문 파일을 입력하고 "복호화" 버튼을 누르면 암호문에 적용된 공개키 알고리즘을 확인할 수 있어 그에 대응하는 비밀키를 사용한다. 또한 복호화 과정이 성공적으로 수행되면 암호문에 포함된 해쉬값(Hash Value)과 복호화 과정에서 생성된 해쉬값의 동일여부를 눈으로 비교하여 복호화 과정이 올바르게 수행되었는지의 여부를 확인할 수 있다.

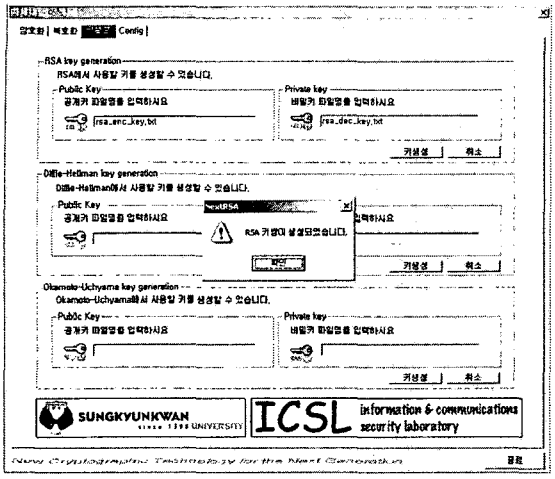
복호화 과정에서 암호문에 사용된 공개키 알고리즘, 대칭키 알고리즘, 해쉬 알고리즘, 송신자의 ID 확인할 수 있으며, 원래의 평문의 파일 이름(File name) 또는 다른 파일 이름으로 복호화할 수 있다.



[그림 5] 복호화 과정

3.3 키 생성 모듈

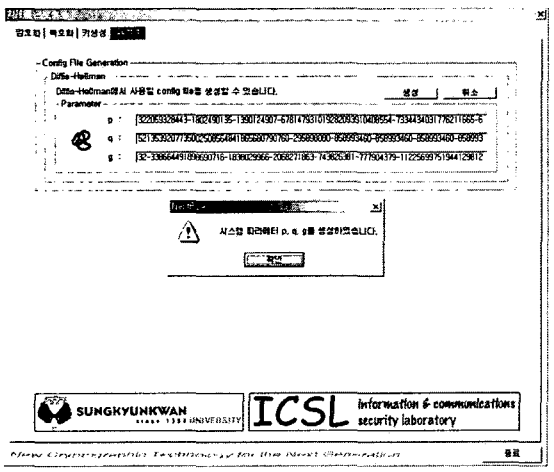
키 생성 모듈은 [그림 6]과 공개키 알고리즘에 사용될 공개키와 비밀키 쌍을 생성하게 된다. 특히, 공개키 알고리즘으로 Diffie-Hellman의 키 전송 알고리즘이 사용되는 경우, 사전에 생성된 도메인 파리메터(Domain parameter)를 이용하여 공개키와 비밀키 쌍을 생성하게 된다. RSA와 Diffie-Hellman의 공개키 알고리즘의 경우 1024 비트(bits) 생성하고 Okamoto-Uchiyama의 공개키 알고리즘의 경우 1023 비트의 키를 생성한다.



[그림 6] 키 생성 과정

3.4 시스템 파라메터 생성 모듈

시스템 파리메터 생성 모듈에서는 Diffie-Hellman의 키 전송 알고리즘이 공개키 알고리즘이 사용되는 경우 필요한 도메인 파라메터로 사용될 p,q,g 를 생성한다. 이 파라메터는 어플리케이션을 사용하는 모든 사용자들이 공유하게 된다.



[그림 7] 시스템 파라메터 생성 과정

3.5 구현한 어플리케이션의 특징

본 논문에서 구현한 메일 어플리케이션은 다음과 같은 특징을 갖는다.

- 안전성이 증명 가능한 암호 시스템 이용
- 공개키 알고리즘, 대칭키 알고리즘, 해쉬 알고리즘의 선택적 사용
- 대칭키 알고리즘의 동작 모드의 선택 가능
- 복호화시 적절한 키 사용 유도
- 일반 E-mail 어플리케이션과 연동 가능
- 일반 E-mail에서 제공하는 모든 기능 사용 가능

4. 결론

본 논문에서는 표준모델에서 안전성이 증명 가능한 공개키 암호 시스템을 실생활에서 활발하게 이용되는 E-mail시스템과 연동하여 간단하게 사용할 수 있도록 어플리케이션을 구현하였다. 이를 구현한 결과를 토대로 하여 상용망에서의 연동 실험을 지속적으로 수행함으로써 상용망에 대한 적응성을 검토하여 이에 대한 보완을 지속적으로 해 나가야 할 것이다. 또한 날로 증가하는 무선을 통한 데이터의 송/수신을 고려하여 본 논문에서 구현에 이용된 암호시스템의 무선으로의 이식성을 고려해야 할 것으로 생각된다.

앞으로 이 어플리케이션을 공개키 기반구조(Public Key Infrastructuer)에서 실제 사용함에 있어 사용자들 간의 공개키를 어떠한 방식으로 공유할 것인가에 대한 연구와 실제 암호문을 주고 받고자 하는 사용자를 확인할 수 있는 사용자 인증(User Authentication)의 문제점에 대해 더 많은 연구가 필요하게 될 것이다.

참고문헌

[1] Diffie, W., and Hellman, M. "New Directions in cryptography", IEEE Trans, Inform, Theory IT-22, pp.644-654, Nov 1976

[2] S. Goldwasser and S. Macali, "Probabilistic Encryption", Journal of Computer and System Science, vol 28(2):270-299, April 1984

[3] D. Dolev, C. Dwork and M. Naor "Non-Malleable Cryptography", Proc. of the 23rd STOC, ACM Press, New York, 1991

[4] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway , "Relations Among Notions of Security for Public-Key Encryption Schemes", Proc. of Crypto 98, LNCS 1462, pp. 26-45. 1998

[5] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems", Commun, ACM, vol. 21, no. 2, pp. 120-126, Feb, 1978

[6] T. Okamoto and S. Uchiyama, "A New Public-Key Cryptosystem as Secure as Factoring", Proc. of Eurocrypt 98, LNCS 1403, pp.308-318, 1998

[7] 최승복, 오수현, 원동호, "표준모델에서 안전성이 증명 가능한 공개키 암호 시스템을 구성하는 일반적인 방법", 정보처리학회 춘계 학술발표논문집 제 8 권 제 1 호, pp. 403-406, 2001. 4

[8] M. Welschenbach, "Cryptography in C and C++", Apress, 2002.

[9] 성균관대학교, "암호키 관리 기반구조 구축을 위한 KMI 라이브러리 개발", 한국정보보호진흥원, 2000. 12