전자봉투를 제거한 ASET(Advanced SET) 프로토콜

양승해* ,신대원**, 이병관***
*관동대학교 전자계산공학과
*관동대학교 전자계산공학과
**관동대학교 컴퓨터공학과
e-mail:yang7177@kwandong.ac.kr
sdw1951@hanmail.net
bklee@kwandong.ac.kr

ASET(Advanced SET) Protocol without Digital Envelope

Seung-Hae Yang* , Dae-Won Shin**, Byung-Kwan Lee***
*Dept of Computer Science, Kwandong University
**Dept of Computer Science, Kwandong University
***Dept of Computer Engineering, Kwandong University

요 약

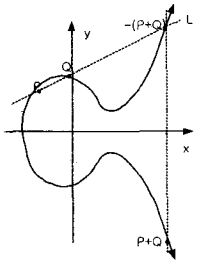
전자상거래 일반적인 구조인 SET(Secure Electronic Transaction)프로토콜은 비밀키 알고리즘의 DES(Data Encryption Standard), 공개키 알고리즘의 RSA(Rivest, Shamir, Adleman), 메시지 다이제스트의 SHA-1를 사용하고 있다. 본 논문에서는 비밀키 알고리즘의 DES를 이용하여 수신측에 전송하는 과정을 3BC알고리즘으로 대체함으로서 생략하였고, 공개키 알고리즘의 요소를 ECC(Elliptic Curve Cryptosystem)알고리즘을 사용하였다. 또한 전자서명을 위한 방법은 Double Signature를 사용하여 SET프로토콜에서 DES의 전송을 위한 전자봉투를 삭제한 결과 수행시간의 단축과 보안의 강도를 강화시켰다.

1. 서론

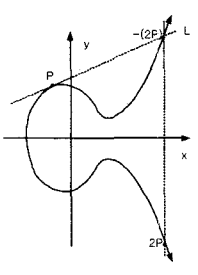
초창기에 인터넷은 특정한 연구기관인 미국 국방성의ARPA(Advanced Research Project Agency)에 의하여 ARPANET이 기원이 되었다. 지금까지의 전자상거래는 일정한 웹 상태에서 이루어지는 것이 보통이었지만, 추후에는 이러한 형식을 벗어나 이동통신과 웹간에 정보를 주고받음으로서 보다 광범위한 자료를 공유할 수 있을 것이다. 전자상거래의 상거래 방식으로 사용되는 프로토콜에는 SET(Secure Electronic Transaction)이 있는데, SET프로토콜이 갖추어야 하는 조건으로는 기밀성, 메시지 무결성, 부인방지, 인증서비스 등이 있다.

2. ECC 알고리즘

ECC(Elliptic Curve Cryptosystem)는 1985년 N.Kobitz와 V.Miller에의해 제안되었다. RSA와 이산대수처럼 일반적으로 사용되는 공개키 보안 시스템과 비교해 볼 때, ECC는 좀 더 작은 키 크기, 그에 따라 더 나은 보안 그리고 더 작은 하드웨어 구현에 이롭다. 이러한 타원곡선은 유한체 상에 정의된 타원곡선에 대하여 타원곡선군은 3차 방정식을 만족하는 순서쌍들과 무한점을 포함한 집합을 말한다. 두 점의 덧셈군을 계산하는 방식으로 두 가지이다. 동일한 점 즉, $P = Q$의 경우와 점 $P \neq Q$인 경우이다. [그림 2-1]은 $P \neq Q$인 경우이고, [그림 2-2]는 $P = Q$인 경우이다.



[그림 2-1] P≠Q의 ECC



[그림 2-2] P=Q의 ECC

공개키 암호화 알고리즘의 키의 생성과정은 아래와 같다.

① [수신자] 소수 p 선택

② [수신자] EC 고정을 위한 a, b을 선택한다.

$y^2 = x^3 + bx + c$

③ [수신자] 무수히 많은 곡선상의 점들 중에서 임의로 초기점 P를 선택한다.

④ [수신자] 정수 k_r 선택하여 개인키로 정한다.

⑤ [수신자] k_rp 값을 addition하여 [그림 2-3]과 같은 방법으로 계산하여 송신자 공개키로 사용한다.

P = Q인 경우	P ≠ Q인 경우
P+P=Q = 2P, P=(x_1,y_1)	P=(x_1,y_1), Q=(x_2,y_2)
2P=Q=(x_3,y_3)	P+Q=(x_3,y_3)
$x_3 = L^2 - 2x_1$	$x_3 = L^2 - x_1 - x_2$
$y_3 = L(x_1 - x_3) - y_1$	$y_3 = L(x_1 - x_3) - y_1$
$L = \frac{3x_1^2 + b}{2y_1}$	$L = \frac{y_2 - y_1}{x_2 - x_1}$

[그림 2-3] ECC Function

⑥ [수신자] p, b, c, P, k_rP를 전송한다.

⑦ [송신자] p, b, c, P, k_rP를 수신한다.

⑧ [송신자] 임의 정수 k_s를 선택하여 수신자의 개인키로 보관한다.

⑨ [송신자] k_sP를 addition 연산하여 수신자의 공개키로 한다.

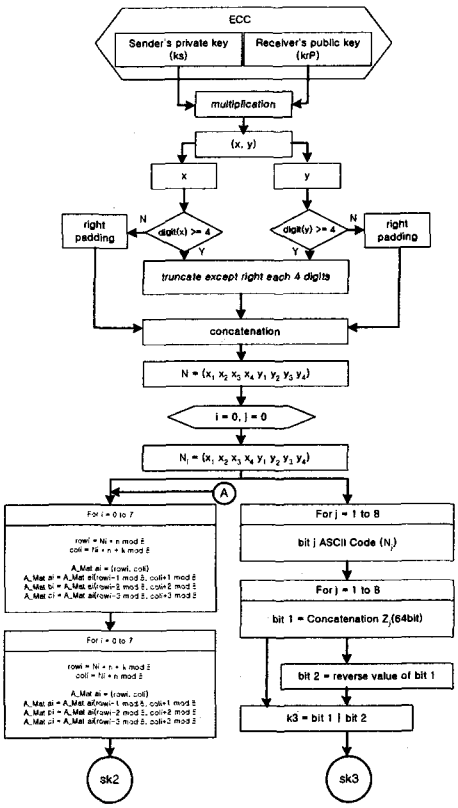
위의 과정의 결과 송, 수신자간의 공유비밀키 즉, 수신측의 공유비밀키인 kr(ksP)와 송신측의 공유비밀키인 ks(krP)를 생성한다. 이후에 각각의 공유비밀키는 3BC알고리즘의 원형키에 사용이된다.

3. 3BC 알고리즘

기존의 SET 프로토콜은 전자서명, 데이터 암호화, 전자봉투를 이용하여 송신측의 정보를 수신측에게 전송을하고, 수신측에서는 각각을 복호화 알고리즘을 이용하여 송신측의 원문을 받아볼 수 있었다. 본 논문에서 제안한 3BC 알고리즘은 ECC알고리즘에서 좌표의 값을 각각 일정한 비트로 잘라내어 송, 수신자간의 공유비밀키의 값으로 사용한다. 따라서 기존의 SET 프로토콜에서 대칭키를 수신측에 전송하기위한 전자봉투를 삭제함으로 프로토콜 전체적인 수행시간을 단축시켰을뿐만 아니라 암호화 강도 또한 증가시켰다. 전자 봉투를 삭제하게 되면, 대칭키를 전송할 필요성이 없어 키의 분실 위험이 없고, 암호화 과정의 일부분이 감소하며, 통신 트래픽도 감소되고 이중 서명의 절차도 간소화 될 수 있으므로 기존의 SET보다 성능 향상을 기대할 수 있다.

3.1 3BC 암, 복호화키 생성알고리즘

3BC 알고리즘은 크게 키 생성부, 데이터 암호화, 데이터 복호화로 구성된다. 키 생성부는 데이터를 암호화하기 위해 ECC 알고리즘의 공유비밀키를 이용하여 원형키를 생성한 후, 블록간의 바이트교환과 bit-xor를 수행하기 위한 Key로 사용된다. [그림 3-1]은 Key Generation부분을 표현한 것이다.



[그림 3-1] Key Generation

Key Generation의 수행과정을 보면 다음과 같다.

① 송, 수신측의 공유비밀키를 ECC알고리즘을 이용하여 multiplication한다.

② 좌표의 x, y의 값을 오른쪽에서 각각 4bit의 크기로 잘라낸다. 만약 좌표의 값이 4bit이하일 때에는 나머지 bit를 0으로 패딩한다.

③ 과정 ②에서 각각의 4bit를 연접(concatenation)하

여 8bit의 원형키로 이용한다.

④ 원형키를 이용하여 블록간의 바이트교환, bit-xor의 키로 사용한다.

⑤ 바이트 교환 : 과정 ④에서 row(Ni * n mod 8), col(Ni * n + k mod 8)를 계산한다.

⑥ 바이트 교환 : 과정 ⑤에서 나온 결과에 row(Ni * n + k mod 8), col(Ni * n mod 8)를 계산한다.

⑦ 바이트 교환 : 과정⑥의 값을 블록간의 바이트 교환키로 정한다.

⑧ bit_xor : 과정④에서 8bit를 각각 ASCII code로 변환한 후 이진수로 출력시킨다.

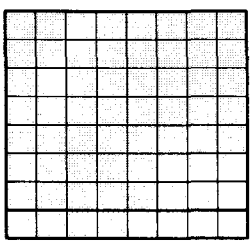
⑨ bit_xor : 과정⑧의 결과에서 각각의 bit열을 reverse한다.

⑩ bit_xor : 과정⑧과 과정 ⑨의 bit열을 xor연산하여 bit-xor키로 정한다.

3.2 3BC 암호화 알고리즘

3BC알고리즘의 블록의 구성을 보면 [그림 3-2]와 같다. 블록이란 64byte의 8×8행렬의 그룹을 의미한다. 먼저 7행까지의 블록인 56byte는 원문데이터가 입력되고, 마지막행의 1~4열은 4byte의 블록의 순서 번호가 입력되는 공간이고, 마지막 행의 5~8열 부분은 4byte의 데이터 교환번호가 입력된다.

암호화 과정은 크게 두 가지로 구분할 수 있다. 먼저 평문으로 구성되어있는 블록들의 쌍간에 동일한 좌표상의 두 블록의 데이터를 서로 교환을 해주는 바이트 교환과 각 행에 해당하는 바이트 교환된 문자를 이진수로 변환하여 공유비밀키를 이용하여 비트간의 xor연산을 한다.



[그림 3-2] 블록의 구성

다음은 데이터를 암호화하는 과정을 설명한 것이다.

① 원문의 데이터를 56바이트의 데이터 영역에 입력한다.

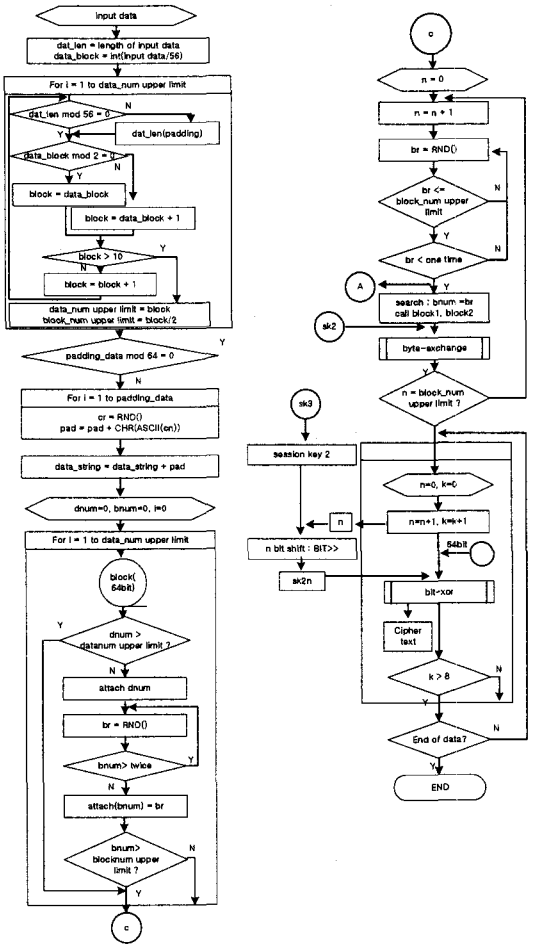
② 블록의 번호를 순서대로 부여하고, 데이터 교환번호를 한 쌍이 되도록 같은 번호를 랜덤하게 지정한다.

③ 3.1절의 키 생성 알고리즘의 과정⑦에 의해서 데이터 교환번호가 같은 쌍간의 블록간에 데이터를 교환한다.

④ 3.1절의 키 생성 알고리즘의 과정⑩에 의해서 블록의 행 단위로 bit-xor시킨다.

⑤ 암호화된 암호화 데이터 순서로 수신측으로 전송한다.

다음의 [그림 3-3]은 데이터를 암호화하는 과정을 보인 것이다.



[그림 3-3] Data Encryption

본 논문에서 제안한 3BC 알고리즘은 암호의 강도를 강화시키는 방법으로 원문데이터가 10개의 블록 이하일 때에는 랜덤하게 10개의 블록까지 만들어줌으로써 블록간의 바이트 교환의 결과로 작은 양의 원문이라도 쉽게 침략자에 의해 노출되지 않게 지정

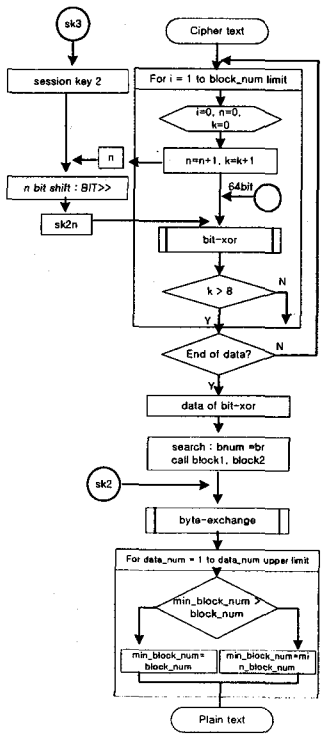
하였다.

3.3 3BC 복호화 알고리즘

암호화 과정에서 전송된 비트열들을 조합하여 블록을 만든 후 각 비트에 해당하는 16진수의 문자로 치환했을 때 당연히 원문과는 다른 문자를 보일 것이다. 복호화 과정의 경우는 다음과 같은 순서를 따른다.

① 송신측으로부터 암호화된 문자열을 입력받는다.
② 3.1절의 키 생성 알고리즘의 과정⑩에 의해서 블록의 행 단위로 bit-xor시킨다.
③ 3.1절의 키 생성 알고리즘의 과정⑦에 의해서 데이터 교환번호가 같은 쌍간의 블록간에 데이터를 교환한다.
④ 블록번호의 순서대로 블록을 나열한다.
⑤ 56바이트의 데이터영역을 맞추기 위해 패딩한 자료를 삭제시킨다.
⑥ 1행부터 차례로 문자들을 정렬하여 평문을 찾아낸다.

3BC 알고리즘의 복호화 과정을 [그림 3-4]에서 볼 수 있다.



[그림 3-4] Data Decryption

4. Double Signature

전자서명 알고리즘으로 본 논문에서는 Double Signature를 제안하였다. 전자서명이란 송신측의 신원을 보장하기 위한 인증의 단계로 원문을 Hash알고리즘을 이용하여 MD(Message Digest)한 결과인 MD(s)와 원문을 수신측으로 전송함으로서 수신측에서는 이 두 가지 자료를 모두 받아 원문을 또다시 수신측에서 Hash알고리즘을 이용하여 MD(r)를 만든다. 다음으로 송신측에서 전송한 MD(s)와 수신측에서 송신측의 원문을 이용하여 Hash한 MD(r)을 비교하여 일치하게되면 송신측의 문서에 대해서 무결성(integrity)을 인정받는다.

본 논문에서는 기존의 SET프로토콜의 DES를 3BC 알고리즘을 이용하여 작성함으로서 수행시간의 단축과 색인표의 공개될 위험요소가 없기 때문에 보안의 강도에서도 더욱 강화된 결과를 기대할 수 있다.

5. 결론

본 논문은 기존의 SET 프로토콜의 과정에서 대칭키인 DES를 전송하기 위한 과정인 전자봉투를 생략하므로 전자봉투의 생성하는 시간의 단축과 DES알고리즘의 단점인 색인표의 도난으로 인한 보안의 위험성, 기본 16라운드의 암호화 과정을 없앰으로서 수행시간의 단축의 장점을 가지고 있다. 또한, 기존의 SET에서 사용되는 RSA 대신에 ECC를 사용함으로서 보안을 강화하였으며, 3BC의 블록간 문자 교환과 비트 연산 알고리즘을 통해 속도의 증가를 기대할 수 있으며 중도에 탈취 당하더라도 평문이 암호화 과정 중에 56byte 단위로 연속성 없이 섞여 있고, 이어 블록간에 문자 교환이 이루어지고, 비트 연산을 통해 암호화되어 개인키의 보관·관리에 문제가 없다면 해독될 가능성은 희박하다.

참고문헌

[1] In-sock, cho "ASEP(Advanced Secure Electronic Payment) Protocol Design", ICIS, 2002. 8, second, pp.366~372.
[2] 정은희, "ECC를 이용한 SSET 프로토콜 설계", 한국정보과학회, 2002. 10. 26, 29권 2호, pp.673~675.
[3] 양승해, "전자신용카드설계를위한ECC알고리즘", 2002.