論文 85-22-3-2

컴퓨터 네트워크의 데이타 보호방식

(A Method for Data Security in Computer Network)

柳 守 恒*, 崔 炳 旭*

(S. H. Ryu and B. U. Choi)

要 約

本 論文에서는 컴퓨터 네트워크 上에서 傳送되는 데이타 또는 多數利用者 시스템(multi-user system)에서 file을 보호하기 위해 이용되고 있는 cryptography에 對하여 論한다. 本 system은 conventional cryptography의 키를 public key cryptography로 관리함으로서 處理速度가 빠르고 키의 관리가 용이하며 새로운 인증자 函數에 의해 확실한 서명문을 얻을 수 있다.

Abstract

In this paper, we describes a cryptography, which is a useful method for data security in computer network and file protection on multi-user operating system. This system manages the keys of conventional cryptography with public key cryptography. As a result, we can obtain high speed encryption, easy manipulation in key management and signatured text by new authentication.

Ⅰ. 序 論

中央의 컴퓨터센터(single computer system)의 性能에 의해 모든 機能이 좌우되던 第3世代 컴퓨터 시스템은 性能의 한계와 독자적인 處理能力의 부족 뿐만 아니라 시스템의 down에 대한 處理對策등의 문제점을 해결하고자 分散된 컴퓨터 시스템에 네트워크를 形成하게 되었으며, 방대한 양의 情報를 보다 효율적으로 관리하고자 데이타 베이스에 의해 데이타를 共有하게 되었다. 한편, O.A의 발달은 전자우편 시스템(electronic mail system)과 on-line system을 확대시켰고, 앞으로 이와같은 디지탈 通信은 더 많이 사용될 것이다. 이렇듯 컴퓨터 네트워크 上에서 傳送되는 데이타는 送受信者가 알지 못하는 사이에 情報가 복사 또는 날조될 수 있으므로 네트워크 채널을 통과하는 데이타를 보호하는 data security는 매우 중요하고도 필수적인 문제로 대두되고 있다.

또한, 네트워크 뿐만 아니라 다수 이용자의 시스템의 O.S에서도 많은 수의 이용자가 data file을 共有함에 따라, 특정 user에 대한 데이타 file의 보호 관리가 필요할 경우가 있다. 현재 다수 이용자 시스템에서는 password에 의해 file을 보호하고 있으나 password란 file관리를 시스템 자체가 하기 때문에 시스템을 잘 아는 특정인에 의해 데이타 보호가 파괴될 수 있다. 또, 多方面으로 研究되고 있는 綜合情報通信網(ISDN)이 구축될 경우 컴퓨터通信(compunication)에 의한 service 업무는 계속 확대되어 사용자 또는 계약자의 신분증명 및 내영증명이 절대적으로 필요하게 될 것이다. 이와같이 데이타를 보호하고 관리하는 방법으로 cryptography가 이용될 수 있다. 이러한 cryptography는 크게 conventional cryptography system(이하 CCS로 表記함)과 public key cryptography system(이하 PKS로 表記함)으로 나눌 수 있고 CCS와 PKS는 處理速度와 安全性에서 서로 相反

*正會員, 漢陽大學校 電子通信工學科

(Dept. of Electro-Communications Hanyang Univ.)

接受日字 : 1984年 8月 31日

※ 본 연구의 일부는 韓國科學財團 82年度 후반기 연구비 지원에 의해 이루어짐.)

되는 장단점을 가지고 있다. 이에 筆者는 이들 두 시스템의 문제점을 改善하는 방법에 대하여 이미 발표한 바 있다.[1,2]

따라서 本 論文에서는 문헌[1][2]를 보완하여 PKS 방식의 대표적인 방법으로 安全性이 확실히 보장되고 있는 RSA法[3]과 같은 정도의 安全性을 가지며, CCS 방식의 대표적인 방법인 DES法[4]과 같은 정도의 處理速度를 갖고, authentication[5]에 의한 digital 署命과 master key[6]의 사용이 가능한 cryptography system(가칭 : CAP system)에 대하여 論한다.

Ⅱ. Cryptosystem의 問題點

1. Algorithm과 處理速度

CCS의 경우 encryption키 벡터 $\vec{ke}$와 decryption키 벡터 $\vec{kd}$는 같으며 식(1)과 같다.

$$\vec{Ke}=\vec{Kd}=\vec{K} \quad (1)$$

Message vector를 $\vec{M}$으로 하고 encryption function을 F로 하는 ciphertext vector $\vec{C}$는 식(2)와 같고

$$\vec{C}=F_{\vec{K}}(\vec{M}) \quad (2)$$

Decryption function을 F^{-1}로 하면 $\vec{c}$로 부터 식(3)과 같이 $\vec{M}$을 얻을 수 있다.

$$F_{\vec{K}}^{-1}(\vec{c})=F_{\vec{K}}^{-1}(F_{\vec{K}}(\vec{M}))=\vec{M} \quad (3)$$

CCS의 경우 8bit microprocessor level에서 400 Kbps~14Mbps까지의 처리량을 갖는 DES chip이 상품화 되고 있을 만큼 encrption(decrption) 속도가 매우 빨라 실제 데이타 通信에서의 傳送속도에는 영향을 주지 않는다. 그러나 채널에서 傳送되는 message를 도청당했을 때 시스템의 알고리즘이 강력하지 못할 경우, 해독이 용이해진다. 또, 사용되는 키가 같으므로 $\vec{k}$는 random number processor에 의해 선택되어야 하며, 키를 傳送할 때는 완벽하게 보호된 채널을 이용해야 한다.[6] 따라서 CCS를 이용하여 데이타를 안전히 傳送하기 위해서는 첫째, 알고리즘이 강력해야 하고, 둘째, 키의 선택, 관리 및 분배에 세심한 주의를 기울여야 하며, 셋째, 受信者에 의해 送信者의 message가 충분히 보호되어야 한다.

한편 PKS는 공개키 K_p와 비밀키 K_s가 存在하고, 이들은 K_s에서 K_p를 구하기는 쉬우나 K_p로 부터 K_s를 구하기는 거의 불가능할 정도의 계산량을 수반하는 單方向函數(one way function)[7]의 관계에 있다. 따라서 PKS는 이러한 單方向函數의 계산량에 의해 安全性을 보장받고 있다. PKS에서 encryption(decryption)은, 傳送하고자 하는 message를 m, encryption과 decryption과정을 각각 E, D로 ciphertext를 C로 하면, 식(4), (5)와 같이 되고, 식(6)을 만족할 경우 서명문을 얻을 수 있다.

$$E_{ke}(m)=C \quad (4)$$

$$D_{kd}(c)=D_{kd}(E_{ke}(m))=m \quad (5)$$

$$D(E)=E(D)=1 \quad (6)$$

이와같이 키가 ke와 kd로 분리되어 있으므로 키를 傳送할 필요가 없으며 送受信키가 다르므로 데이타를 완벽하게 보호할 수 있다. 그러나 계산량이 너무 많으므로 처리 시간이 매우 길어진다.[3]

이와같이 각 시스템은 相反되는 장단점을 가지고 서로 발전되어, 알고리즘이 상당히 강력해졌음에도 불구하고 근본적인 문제점을 모두 해결할 수는 없었다.

2. Digital Signature and Authentication

Digital信號에 의해 중요한 서신을 送受信하는 경우 送信者 A가 보낸 서신을 받은 受信者 B는 그 message가 확실히 A가 보냈는지, 또는 그 내용이 진실한 것인지를 알 수가 없다. 이러한 문제를 해결하는 방법으로는 digital signature와 authentication이 이용될 수 있다. authentication은 주로 conventional한 방법에서 사용되는 것으로 서명문의 일종인 authenticator를 만들어 message의 내용을 확인하는 것이며 그 개략도를 그림 1에 보인다.[8]

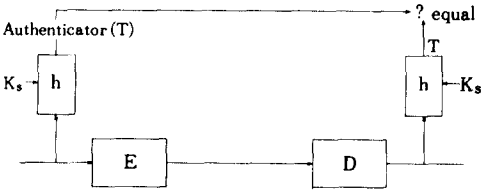

그림 1. CCS에서의 인증자 비교법
Fig. 1. Authenticator verification method using CCS.

여기서 function h는 message로 부터 authenticator를 만들기는 쉬우나 authenticator로 부터 message를 복원하기는 매우 어려운 單方向函數를 이용한다. 그러나 이 방법의 경우 受信者 B가 임의로 authenticator를 만들 수 있으므로 후에 論爭의 여지가 있게 된다.

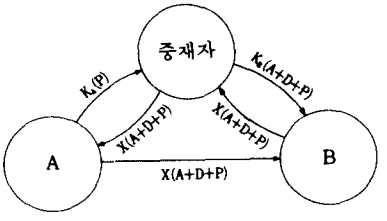

그림 2. 증재자를 통한 서명법
Fig. 2. Secure message transmission between strangers using CCS with signature.

따라서 conventional한 방법의 경우 믿을 만한 중재자를 通한 간접서명법이 이용되고 있으며 그림 2와 같다.[9]

그러나 진정한 의미의 署命은 일부 public key cryptosystem에서 가능한 技法이다. 그중 가장 대표적인 RSA法에서는 그림 3 과 같이 표시된다.

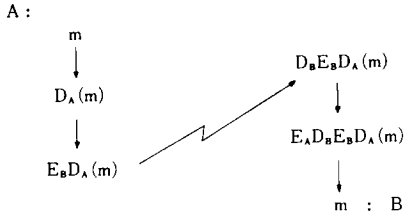

그림 3. RSA 법을 이용한 서명법
Fig. 3. Digital signature using RSA.

이 경우 A의 비밀키 없이는 署命文 $D_A(m)$을 만들 수 없으므로, 완벽한 署命文을 얻을 수 있다. 그러나 계산량이 너무 많다는 문제점이 남게 된다. 이러한 署命 및 認證에 대한 비교를 표 1, 2에 보인다.[2]

표 1. 인종의 비교
Table 1. Comparision of authentication.

	CCS	PKS
인 증	가능	사용치 않음
간접서명	중재자를 통하여 가능	사용치 않음
직접서명	불가능	가능

표 2. 처리 효율
Table 2. Processing efficiency.

	인 증	간접서명	직접서명
처 리 량	M+h	2M+S	mlEl
처리속도	상	상	하
전 송 량	M+h(중)	M+S(중)	m(상)
Redundancy	중	*하	상
신 뢰 도	**하	상	상

m : message
h : 인증자
s : 중재자의 sign
E : encryption하는데 필요한 처리량
* : 중재자에 의한 서명문이 추가
**: 수신자에 의한 조작이 가능

Ⅲ. High Speed Encryption and Authentication System

이와같이 encryption과정 뿐만 아니라 署命 및 認證에 있어서도 相反되는 장단점이 存在하기 때문에 두개의 시스템을 組合 할 경우 매우 큰 效果를 얻을 수 있다. 本 시스템은 CCS와 PKS에서 대표적인 방법으로 알려진 DES法과 RSA法을 이용하고 있다. 그에 대한 블럭도를 그림 4 에 보인다.

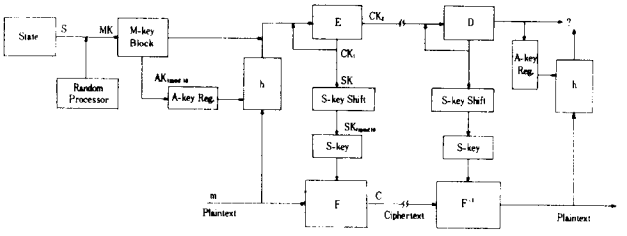

그림 4. CAP Crypto system의 개략도
Fig. 4. Block diagram of CAP crypto system.

CAP crypto-system의 암호과정은 mother key (이하 MK로 표기) 생성부, son key (이하, SK로 표기) 생성부와 암호화 부분으로 구성된다. MK는 state와 random processor에 의해 생성되며, 여기서 state는 plaintext에서 사용되는 문자의 종류와 code 方式에 의해 결정되는 표준화된 정보를 선택하는 것으로 최상위 bit가 0으로 set되어 있는 16bit의 block을 0~39개까지 선택한다. 이러한 state는 random processor에 의해 생성된 최상위 bit가 1로 set된 random 16bit block의 무의미한 정보를 삽입하여 640bit의 random vector를 생성하며, 이 640bit가 MK로 된다. SK 생성부는 RSA와 shift register로 구성되며 MK는 受信者의 공개키에 의해 encryption되어 CK1을 생성하고 CK1의 하위 640bit가 SK shift register에 저장되며 CK1을 재차 encryption하여 얻은 CK2를 전송한다. CK2를 받은 수신자는 역과정에 의해 MK, SK state를 얻을 수 있다. SK shift register에 저장된 SK는 10개의 64bit키 $SK_i(SK_1, SK_2, \cdots SK_9, SK_0)$로 분리되어 SK register에 1개의 64bit block씩 선택된다. 암호화 부분은 SK register와 DES로 구성되며 n번째 plaintext block P_n은 i=n mod10으로 선택되는 SK_i에 의해 encryption되는 shift key方式에 의해 처리되며 decryption도 같다.

인증자 함수는 서명문을 얻을 때 사용하는 것으로 서명문을 필요로 하지 않는 message에 대하여는 이 과정을 생략 할 수도 있다. 인증자 함수 과정은 64bit의 authentication key (AK) register와 인증자 함수 h로 구성된다.

AK는 MK block register에서 64bit씩 shift되어 선택된다. h함수는 vector A와 vector B를 연결하는 함수 CONCAT(A, B)에 의해 다음과 같이 정의된다.

```
PROCEDURE h(Pi);

begin
  RT[0]:=″;
  for i:=1 to 128 do
    RT[0]:=CONCAT(RT[0],'0');
  for i:=1 to N do
    begin
      T[i]:=EXOR(RT[i-1],
            CONCAT(Pi, AK[i mod10]));
      RT[i]:=REVERSE(T[i])
    end;
  T:=T[N]
end;
```

(여기서, reverse는 128bit의 T[i]를 역순으로 배열하는 함수이고, EXOR은 exclusive OR의 연산을 하는 함수이다).

이렇게 생성된 T는 모든 message vector에 대하여 고유하게 만들어지므로 message없이는 생성이 불가능하게 되고, 함수 h는 일종의 one way function으로 작용하게 된다. 또한 T를 송신자의 비밀 key로 decryption하고 수신자의 공개 key로 encryption한 $E_BD_A(T)$를 수신자에게 전송하여 주고, 이를 받은 수신자는 자신의 비밀키로 decryption하여 $D_A(T)$를 얻을 수 있으며 이것은 송신자의 비밀키 없이는 생성이 불가능하므로 확실한 서명문이 된다. 이 서명문은 송신자의 공개키로 encryption하여 T를 얻을 수 있다. 시스템의 처리 과정은 다음과 같다.

```
PROCEDURE cap-encryption;

begin
  GET(S); (State initial seed)
  MK:=RANDOM-P(S);
       (Generate mother key from random
        processor)
  CK1:=E(KBe, MK);
  SK:=copy(CK1, length(CK1)-639, 640);
       (Generate son key)
  CK2:=E(KBe, CK1);
  TRANSMEET(B, CK2);
  for i:=1 to 10 do
     SK[i]:=copy(SK, 64*(i-1)+1, 64);
  SK[0]:=SK[10];
  RT[0]:=″;
  for i:=1 TO 128 do
     RT[0]:=concat(RT[0],'0');
  for i:=1 TO N do
     begin
      CT[i]:=FE(SK[i mod 10], P[i]);
      T[i]:=EXOR(RT[i-1], concat(P[i], AK
            [i mod 10]));
      RT[i]:=REVERSE(T[i]);
      TRANSMEET(CT[i])
     end;
  ST:=D(KAd, T[N]);
  TRANSMEET(E(KBe, ST))
END;
```

이상과 같은 CAP system의 encryption 과정에서는 다수의 보조키에 의해 處理速度를 向上시켰고, 동시에 解讀을 곤란하게 하였다. 또, 키의 관리를 PKS에 의해 一元的으로 處理함으로써 key 관리가 용이하고 알고리즘이 강력한 시스템을 講成하였다. 또한 새로운 認證子函數를 도입하여 encryption에 부하를 주지 않고 병렬처리에 의해 모든 message에 대하여 고유한 authenticator를 만들어 認證을 可能하게 한다.

Ⅳ. 速度 比較

DES의 경우 through put이 400K bps~14Mbps로 상품화 되어 있고 RSA의 경우 hardware로 구성할 경우 50K bps로 실현 可能하다고 한다.[11] DES의 경우 64bit씩, RSA의 경우 10^{200}(약 640bit) 정도씩 처리 되므로 1block당 DES와 RSA의 처리량은 (7)式과 같다.

$$\frac{400,000}{64} : \frac{50,000}{640} = 80 : 1 \qquad (7)$$

따라서 DES와 RSA의 처리 시간비는 式(8)과 같다.

$$F(\text{or } F^{-1}) : E(\text{or } D) = 1 : 80 \qquad (8)$$

따라서 x bit의 data를 처리하기 위해서 必要한 時間은 DES의 경우 式(9)와 같고

$$T_D = \frac{x}{64} \times F \qquad (9)$$

RSA를 사용할 경우 式(10),

$$T_R = \frac{x}{640} \times E = 8 \times \left(\frac{x}{64} F\right) \qquad (10)$$

RSA를 통하여 署名을 할 경우 式(11),

$$T_{RS} = \frac{x}{640}(E+D) = 16 \times \left(\frac{x}{64} F\right) \qquad (11)$$

CAP를 통하여 署名할 경우 式(12)와 같다.

$$T_C = 2E + (E+D) + \frac{x}{64} F = \left(320 + \frac{x}{64}\right) \qquad (12)$$

따라서 署名을 하지 않는 RSA와 署名을 하는 CAP system과 속도가 같아지는 x의 값은 식(10), (11), (12) 의해

식(13)과 같이

$$320+\frac{x}{64}=\frac{8}{64}x \qquad (13)$$

x=2926으로 약 3,000bit 이상일 때 부터 그림(5)의 比率로 RSA보다 빨라지며 RSA에 의해 서명을 할 때는 1,500bit 이상에서부터 빨라짐을 알 수 있다.

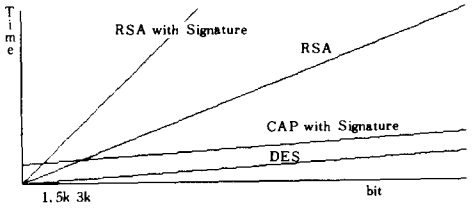

그림 5. 처리속도 비교
Fig. 5. Through put ratio.

3,000bit는 375byte로 일반적으로 우리가 처리해야 할 情報는 이것보다 훨씬 많은 data이기 때문에 매우 빠른 속도로 ciphertext와 署名文을 얻을 수 있다.

V. 結 論

이상과 같이 CCS와 PKS를 結合한 CAP system을 構成하였다.

本 시스템은 conventional한 特性에 의하여 encryption 速度를 向上 시켰고, PKS의 特性에 의하여 키의 관리가 용이하다.

또, 새로운 認證子函數를 도입하여 안전한 digital signature를 함으로 身分 및 內容證明이 확실하며, RSA에서와 같은 方法에 의하여 master key를 利用할 수도 있으므로 情報管理가 容易함을 알 수 있다. 또한 DES보다 強力한 ciphertext를 얻을 수 있으므로 network protocol에 本 시스템을 適用할 경우 high level layer에서, plaintext의 state에 對한 情報를 제공한다면, 어느 기종 간에도 通信이 可能하고 處理時間이 빠르므로 채널 便用 時間을 줄일 수 있다.

參 考 文 獻

[1] 유수항, 최병욱, "Computer Network의 Data 보호방식 Ⅰ", 대한전자공학회 추계종합학술대회 논문집, vol. 6, no. 2, 1983.

[2] 유수항, 최병욱, "Comuter Network의 Data 보호방식Ⅱ", 대한전자공학회 하계종합학술대회 논문집, vol. 7, no. 1, 1984.

[3] R. Rivest, A. Shamir and L. Adlman: "A method for obtaining digital signatures and public key cryptosystem," *C-ACM*, vol. 21, no. 2, pp. 120-128, Feb. 1978.

[4] NBS: *Data Encryption Standard Federal Information Processing Standards.* Pub 1. 46,1977.

[5] W. Diffie and M.E. Hellman: "Privacy and authentication: An introduction to cryptography," *Proceeding of the IEEE*, vol. 67, no.3, Mar. 1979.

[6] 고윤석, "Data통신에 있어서 안전성에 관한 연구", 한양대학교 석사학위 논문, 1983.

[7] R.C. Mercle and M.E. Hellman: "Hiding Information and Signatures in Trapdoor Knapsacks," *IEEE Trans. Inf. Theory* vol. IT-24 no. 5, Sept. 1978.

[8] D.W. Davis: *"Applying the RSA Digital Signature to Electronic Mail,* IEEE Computer, Feb. 1983.

[9] A.S. Tanenbaum; *Computer Network.* p-H Pub. pp. 386-439.

[10] S. Miyaguchi : "Fast encryption algirithm for the RSA cryptographic system" *IPS of Japan*, vol. 24, no. 6, Nov. 1983.

[11] R. Akiyama and R. Yathubosh : "Trends on the hardware technologies for data encryption," *IPS of Japan,* vol. 25, no. 6, Jun. 1984.