論文 97-22-2-16

RSA 블럭 보호 방법과 그 응용

正會員 박 상 준*, 원 동 호**

New RSA Blocking Method and Its Applications

Sang Joon Park*, Dong Ho Won** *Regular Members*

요 약

RSA 암호는 데이터 암호와 서명을 함께 실현할 수 있는효율적인 공개키 암호 방식이다. 그러나, 많은 사용자들이 하나의 서류에 서명을 하고자 할 경우 사용자들이 사용하는 RSA modulus 크기 차이로 인하여 블럭 보호(blocking) 문제가 발생한다. 본 논문에서는 RSA modulus 값의 크기에 관계없이 입력되는 데이터의 크기에 따라 메시지 블럭의 크기를 재구성하여 서명을 생성하는 새로운 RSA 블럭 보호 방법을 제안하였다. 제안된 블럭 보호 방법은 서명 순서를 제한받지 않는 다중 서명(multisignature) 방식에 응용할 수 있을 뿐 아니라, 다중 서명된 메시지를 비밀리에 수신자에게 전송할 수 있다. 본 서명 방법에 의하여 생성되는 다중 서명은 RSA modulus 비트 사이즈 보다 커진다. 그러나, 확장되는 비트 사이즈는 각 서명자의 RSA modulus 크기에 관계없이 다중 서명에 참여한 서명자의 수 보다 작다.

ABSTRACT

In this paper, we propose a new blocking method in which the size of an encryption block is changed according to the size of a message block. The proposed method can be applied to a multisignature scheme with no restriction of the signing order and a multisignature can be sent secretly to the receiver through RSA encryption. It causes expansion in block size of a multisignuture, but the length of the expanded bits is not greater than the number of signers regardless of the bit lengths of RSA moduli.

I. 서 론

1978년 Rivest, Shamir, Adleman은 인수 분해 문제의 어려움에 근거한 RSA 공개키 암호시스템을 제안하였다. RSA 암호는 데이터 암호와 서명에 적용 가능하며 이때, 생성되는 암호문과 서명문은 입력되는 평문과 동일한 사이즈를 갖는다. 그러나, 데이터 암호와 서명을 동시에 실현하고자 하든가 많은 사용자들이 하나의 서류에 서명하고자 할 경우, 각 사용자들이 사용하는 RSA modulus의 크기 차이로 인하여 블럭

*한국전자통신연구소
**성균관대학교 정보공학과
論文番號:96208-0718
接受日字:1996年 7月 18日

보호 문제(blocking problem)가 발생한다[1].

Kohnfelder는 RSA modulus n의 크기에 따라 암호화 순서를 정하는 방법을 사용함으로서 블럭 보호 문제를 해결할 수 있음을 보였다[2]. 또한, Levine과 Brawley는 모든 사용자들이 같은 비트 사이즈를 갖는 RSA modulus를 사용할 경우 반복적으로 암호화시키는 방법(re-encrypted technique)을 사용하여 블럭 보호 문제를 해결하였다[3].

Itakura와 Nakamura는 RSA에 기반을 둔 다중 서명 기법을 최초로 제안하였다[4]. 다중 서명 기법이란 하나의 서류에 여러명의 서명자가 서명하는 방법을 말한다. 따라서, RSA를 다중 서명에 적용하고자 할 경우 블럭 보호 문제가 발생하게 된다. 그들은 이러한 블럭 보호 문제를 해결하기 위하여 서명자가 갖는 조직 내의 지위에 따라 RSA modulus의 크기를 정하였다. 그러므로, 시스템에 가입하지 않은 제 3자가 포함될 경우 다중 서명이 불가능할 뿐 아니라, Kohnfelder의 방법과 마찬가지로 지위에 따라 서명의 순서가 정해진다.

Okamoto가 제안한 다중 서명 방법에서는[5] 이전 서명자에 의하여 생성된 서명문의 크기가 자신의 RSA modulus 보다 커질 경우 RSA modulus의 비트 사이즈보다 큰 서명문의 상위 비트를 평문에 첨가하고 나머지 하위 비트 정보만을 자신의 비밀키로 서명하여 다음 서명자에게 메시지와 함께 전달한다. Okamoto 방법은 서명자 수와 RSA modulus의 비트 사이즈에 따라 메세지의 크기가 증가하며, 서명 검증시 메시지에 첨가된 중간 서명문의 부분 정보를 다시 복원시켜야 하므로 검증 과정이 복잡하다.

Harn과 Kiesler는 Kohnfelder 방법을 일반화시켜 서명 생성시 비트 확장을 유발하지 않는다중 서명 방법을 제안하였다[2][6]. 예를들어 k명의 사용자가 하나의 서류에 서명 하고자 할 경우 k개의 RSA 키 중에서 가장 작은 RSA modulus를 가진 사용자 부터 차례로 서명을 한다. 따라서, RSA modulus의 크기에 따라 서명의 순서가 정해진다. 그들은 각 사용자가 크기가 서로 다른 두개의 RSA 키를 갖도록 하여 서명문을 수신자에게 비밀리에 보낼 수 있도록 하였다. 이 경우, 시스템에서 정한 임계치(threshold value) 보다 작은 비트 사이즈를 갖는 RSA 키는 서명을 위하여 사용되며 임계치보다 큰 RSA 키는 서명문의 암호화/복호화를 위하여 사용된다[2][6].

본 논문에서는 메세지의 크기에 따라 RSA 암호 블럭의 사이즈가 가변되는 새로운 블럭 보호 방법을제안하고자 한다. 제안된 블럭 보호 방법은 인증 키 분배(Authenticated Key Distribution)와 다중 서명등에 적용 가능하다. 다중 서명에 적용시 사용자 수만큼 서명문의 비트 크기가 증가될 수 있으나 서명자의 수가 적을 경우 확장되는 비트 길이는 무시할 수 있다. 또한, Okamoto의 서명 방법과 달리 확장되는 비트 수가 각 서명자가 갖는 RSA modulus의 크기와 무관하며, 하나의 RSA 키만을 사용하여 서명문을 수신자에게 비밀리에 전달할 수 있다.

본 논문은 모두 6개 절로 구성된다. 2절에서는 Okamoto의 다중 서명 방법을 소개하였으며, 3절에서는 Harn과 Kiesler 서명 방법을 소개하였다. 4절에서는 본 논문에서 제안하는 새로운 블럭 보호 방법을 기술하였으며, 5절에서는 제안된 블럭 보호 방법을 응용한 인증 키 분배 및 다중 서명 방법을 제안하였다. 6절은 본 논문의 결론부이다.

Ⅱ. Okamoto의 다중 서명

Okamoto는 RSA와 같이 암호 함수가 전단사이고 사용자들이 사용하는 암호 함수의 정의역(domain)의 크기가 서로 다를 경우에 적용할 수 있는 다중 서명 방법을 제안하였다[5]. Okamoto 방법은 서명 순서에 제한을 받지 않으며 모든 사용자가 동일한 비트 사이즈를 갖는 RSA modulus를 사용할 경우 다중 서명 과정에서 증가되는 비트 수는 서명자의 수와 같다.

이제 RSA 암호를 이용한 다중 서명의 예를 통하여 Okamoto의 서명 방법을 소개하고자 한다. 설명의 편의성을 위하여 사용자 $U_1, U_2, \ldots, U_k$를 서명자라 하고 U_{k+1}을 다중 서명의 수신자라고 하자. 다음은 본 논문에서 각 사용자의 RSA 키와 다중 서명 과정에서 자주 사용되는 표기이다.

- n_i : 사용자 U_i의 RSA modulus
- (e_i, n_i) : RSA 공개키, (d_i, n_i) : RSA 비밀키 $(e_i \cdot d_i = 1 \bmod \Phi(n_i))$
- $|n_i|$: n_i의 비트 사이즈를 의미
- $A \| B$: A와 B의 비트 연접(concatenation)을 의미
- $h(\cdot)$: 충돌 회피성을 갖는 해쉬 함수

• S_i: 사용자 U_i의 서명문

서명 생성 및 검증 과정

• 서명자 U_1에 의한 서명:

메세지 M에 대한 해쉬값 $h(M)$을 자신의 비밀키로 암호화하여 서명문 $S_1 = h(M)^{d_1} \bmod n_1$을 생성하고 메세지 $M_1 = M \| M_1$과 함께 U_2에게 전달한다 (X_1은 NULL, 즉 $|X_1| = 0$).

• 서명자 U_i에 의한 서명 ($i = 2, ..., k$)

1) 사용자 U_i는 메세지 M과 이전 서명자들 U_1, U_2, ..., U_{i-1}에 의하여 생성된 서명문의 부분 정보 X_1, X_2, ..., X_{i-1}이 첨가된 메세지 $M_{i-1} = M \| X_1 \| X_2 \| ... \| X_{i-1}$와 서명문 S_{i-1}을 수신한다.

2) 만일 $|n_i| > |n_{i-1}|$이면,

$Y_i = S_{i-1}$, X_i = NULL

만일 $|n_i| \le |n_{i-1}|$ 이면,

$$Y_i = S_{i-1} \bmod 2^{|n_i|-1},\ X_i = \left\lfloor \frac{S_{i-1}}{2^{|n_i|-1}} \right\rfloor,\ S_{i-1} = X_i \| Y_i$$

$|n_i| \le |n_{i-1}|$이고, $X_i < 2^{|n_{i-1}|-|n_i|}$이면 X_i의 상위 비트를 '0'으로 padding하여 X_i의 비트 길이를 $|n_{i-1}| - |n_i| + 1$로 만든다. 따라서, $|n_i| > |n_{i-1}|$이면 $|X_i| = 0$이고, $|n_i| \le |n_{i-1}|$이면 $|X_i| = |n_{i-1}| - |n_i| + 1$이 된다.

3) 메세지 M_i와 서명문 S_i를 다음과 같이 생성한다.

$$M_i = M_{i-1} \| X_i,\ S_i = Y_i^{d_i} \bmod n_i$$

4) M_i와 S_i를 다음 사용자 U_{i+1}에게 전달한다.

• 서명 수신자 U_{k+1}에 의한 서명 검증 과정

1) $i = k, k-1, ..., 2$에 대하여, 다음과 같은 방법으로 서명 S_k가 사용자 U_1, U_2, ..., U_k의 서명임을 확인한다.

만일 $|n_{i-1}| < |n_i|$이면,

$S_{i-1} = S_i^{e_i} \bmod n_i$, $X_i = NULL$

만일 $|n_{i-1}| \ge |n_i|$이면,

$S_{i-1} = X_i \| (S_i^{e_i} \bmod n_i)$, $X_i = M_i$의 오른쪽 $|n_{i-1}| - |n_i| + 1$ 비트

2) 마지막으로 S_1이 메세지 M의 해쉬값 $h(M)$에 대한 사용자 U_1의 서명임을 확인하다.

$$h(M) = S_1^{e_1} \bmod n_1$$

Okamoto가 제안한 다중 서명 방법의 주요 특징은 다중 서명 과정에서 메세지 사이즈가 증가된다는 것이다. 예를들어, U_{i-1}의 RSA 키 n_{i-1}의 비트 사이즈가 다음 서명자 U_i의 RSA 키 n_i의 비트 사이즈 보다 크거나 같을 경우, 서명자 U_i는 수신한 메세지 M_{i-1}를 $|X_i| = |n_{i-1}| - |n_i| + 1$ 비트 증가시킨 메세지 $M_i = M_{i-1} \| X_i$를 생성하게 된다. 제안된 방법은 k명의 사용자가 모두 동일한 비트 길이를 갖는 RSA 키를 사용할 경우에는 최대 k 비트의 메세지 확장을 유발시키며, 따라서 비트 확장률은 상대적으로 크지 않다. 그러나, 서명자들이 서로 다른 비트 길이를 갖는 RSA 키를 사용한다면 확장되는 비트의 수는 크게 증가될 수 있다. 예를들어, 6명의 사용자가 다음과 같은 비트 길이를 갖는 RSA 키를 사용한다고 하자.

$$|n_1| = |n_3| = |n_5| = 768,\ |n_2| = |n_4| = |n_6| = 512$$

이 경우, 최종 서명문 S_6의 크기는 512 비트이나, 6명의 서명자에 의하여 메세지 M에 첨가되는 정보 X_1, X_2, X_3, X_4, X_5, X_6의 비트 길이는 다음과 같다.

$$|X_1| = |X_3| = |X_5| = 0,\ |X_2| = |X_4| = |X_6| = 257$$

그러므로, 확장되는 메세지 비트의 길이는 총 $\Sigma_i X_i = 771$ 비트가 된다. Okamoto는 위에서 제안한 다중 서명 방법 이외에 3개의 변형된 다중 서명 방법을 제안하였으나 확장되는 비트 사이즈는 본 절에서 소개한 방법과 유사한 결과를 갖는다.

Ⅲ. Harn-Kiesler의 다중 서명

Harn과 Kiesler는 Kohnfelder 방법을 일반화시켜 서명 생성시 비트 확장을 유발하지 않는 다중 서명 방법을 제안하였다[2][6]. 예를들어 k 명의 사용자가 하나의 서류에 서명 하고자 할 경우 k개의 RSA 키 중에서 가장 작은 RSA modulus를 가진 사용자부터 차례로 서명을 한다. 따라서, RSA modulus의 크기에 따라

서명의 순서가 정해진다. 그들은 각 사용자가 크기가 서로 다른 두개의 RSA 키를 갖도록 하여 서명문을 수신자에게 비밀리에 보낼 수 있도록 하였다. 먼저 시스템의 관리자는 임계치(threshold value) h를 미리 설정한다. 각 사용자 U_i는 서명을 위한 RSA 키와 비밀 통신을 위한 RSA 키를 다음과 같이 생성한다.

- 서명용 RSA 키 : (e_{i1}, n_{i1}), (d_{i1}, n_{i1})
  $e_{i1} \cdot d_{i1} = 1 \bmod \phi(n_{i1})$, $n_{i1} < h$
- 비밀 통신용 RSA 키 : (e_{i2}, n_{i2}), (d_{i2}, n_{i2})
  $e_{i2} \cdot d_{i2} = 1 \bmod \phi(n_{i2})$, $n_{i2} > h$

그러므로, 각 서명자의 어떤 서명용 RSA 키 n_{i1}도 비밀 통신용 RSA 키 n_{j2} 보다 작다.

다중 서명에 앞서 초기 주관자(initiator)는 다중 서명에 참여하는 서명자들의 서명용 RSA 키의 크기를 비교하여 n_{i1}의 크기에 따라 서명의 순서를 정한다. 일반성을 잃지않고 서명에 참여하는 사용자를 U_1, U_2, ..., U_k라하고 이들이 사용하는 서명용 RSA 키가 $n_{11} < n_{21} < \ldots < n_{k1}$라 하자.

- 사용자 U_1에 의한 서명
  1) 메세지 M의 서명 $S_1 = M^{d_{11}} \bmod n_{11}$ 계산
  2) 서명 S_1의 암호문 $C_1 = S_1^{e_{22}} \bmod n_{22}$ 계산
  3) C_1을 다음 서명자 U_2에게 전달
- 사용자 U_i에 의한 서명 (i = 2, 3, .., k)
  1) C_{i-1}를 복호화하여 $S_{i-1} = C_{i-1}^{d_{i1}} \bmod n_{i2}$ 계산
  2) S_{i-1}이 메세지 M에 대한 $U_1, U_2, \ldots, U_{i-1}$의 다중 서명임을 확인
     $S_{j-1} = S_j^{e_{j1}} \bmod n_{j1}$ (j = $i-1, i-2, \ldots, 2$),
     $M = S_1^{e_{11}} \bmod n_{11}$
  3) U_i의 서명 $S_i = S_{i-1}^{d_{i1}} \bmod n_{i1}$ 계산
  4) 서명 S_i의 암호문 $C_i = S_i^{e_{(i+1)2}} \bmod n_{(i+1)2}$ 계산
  5) C_i를 다음 서명자 $U_{(i+1)}$에게 전달
- 수신자 U_{k+1}에 의한 다중 서명 검증
  1) C_k를 복호화하여 $S_k = C_k^{d_{(k+1)2}} \bmod n_{(k+1)2}$ 계산
  2) S_k가 메세지 M에 대한 $U_1, U_2, \ldots, U_k$의 다중 서명임을 확인
     $S_{j-1} = S_j^{e_{j1}} \bmod n_{j1}$ ($j = k, k-1, \ldots, 2$),
     $M = S_1^{e_{11}} \bmod n_{11}$

Harn-Kiesler의 방법은 서명문의 크기가 다중 서명에 참가하는 서명자의 수에 관계없이 비트 확장을 유발하지 않는 장점이 있으나 서명 순서가 제한되며 다중 서명 과정중에 새로운 사용자가 다중 서명에 참여할 수 없는 단점이 있다.

Ⅳ. 새로운 RSA 블럭 보호 방법

본 절에서는 암호화하고자 하는 메세지의 크기에 따라 RSA 암호 블럭의 크기가 가변되는 새로운 블럭 보호 방법을 제안하고자 한다. n을 RSA moduli라하고 e를 $\gcd(e, \phi(n)) = 1$을 만족하는 RSA 공개키라 하자. 또한, 평문 M이 홀수이고 $0 < M < 2^l \cdot n$라 하자. $\phi(2^l \cdot n) = 2^{l-1} \cdot \phi(n)$ 이므로 $\gcd(e, 2^{l-1} \cdot \phi(n)) = 1$이 성립한다. 따라서, $e \cdot d = 1 \bmod 2^{l-1} \cdot \phi(n)$을 만족시키는 d가 존재한다. 그러므로, 홀수인 평문 M은 다음과 같은 관계식을 만족시킨다.

$$M^{e \cdot d} = M^{d \cdot e} = M \bmod 2^l \cdot n$$

위의 관계식에서 암호 블럭 사이즈 $2^l \cdot n$는 l에 따라 블록 사이즈를 증가시킬 수 있으므로 평문 M이 RSA modulus n 보다 큰 값을 갖는다 하여도 암호화/복호화 가능하다. 이때, l 값의 변화에 따라 비밀키 d의 값도 함께 가변된다. $C = M^e \bmod 2^l \cdot n$, $e \cdot d_1 = 1 \bmod 2^{l-1}$이라 하면 다음의 관계식이 성립한다.

$$C \bmod 2^l = M^e \bmod 2^l,\ M \bmod 2^l = C^{d_1} \bmod 2^l$$

d_1의 값은 누구든지 계산할 수 있으므로 비밀키 d를 모르는 사람도 암호문 C로 부터 평문 M의 하위 l 비트를 구할 수 있다. 따라서, 제안된 블럭 보호 방법은 큰 블럭 사이즈를 갖는 평문을 암호화시키는 방법으로 직접 사용할 수 없다. 그러나, 다음의 정리는 암호문 C로 부터 평문 M의 모든 비트 정보를 구한다는 것이 임의의 평문에 대한 RSA 서명을 계산하는 것과 같이 매우 어려운 문제라는 것을 보여준다.

정리1 e, n, l로 부터 $e \cdot d = 1 \bmod \phi(n) \cdot 2^{l-1}$인 비밀키 d를 구할 수 있다면 임의의 메세지 $M(0 < M < n)$에 대한 RSA 서명을 계산할 수 있다.

(증명) $d'=d \bmod \phi(n)$이라 하면 $C=M^d=M^{d'} \bmod n$이고 $e\cdot d'=e\cdot d=1 \bmod \phi(n)$. 따라서, d'은 공개키 (e, n)에 대응되는 RSA 비밀키이며, $C \bmod n$은 메세지 M에 대한 RSA 서명이다. 그러므로, 비밀키 d를 구할수 있다면, 임의의 평문 M에 대한 RSA 서명을 계산할 수 있다. □

정리2 모든 홀수 $M(0<M<2^l\cdot n)$에 대하여 $C^e=M \bmod 2^l\cdot n$인 C를 계산할 수 있다면 임의의 홀수 $M(0<M<n)$에 대한 RSA 서명 $C(C^e=\mathrm{M} \bmod n)$를 계산할 수 있다.

(증명) M을 $0<M<n$이고 홀수인 평문이라 하자. 가정에 의하여 $C^e=M \bmod 2^l\cdot n$인 C를 계산할 수 있다. 따라서, $C^e=M \bmod n$이 성립하며 $C \bmod n$은 M에 대한 RSA 서명이 된다. □

앞의 (정리 1)은 비밀키 d를 구하는 것이 임의의 평문에 대한 RSA 서명을 계산하는 것만큼 어렵다는 것을 의미하며, (정리 2)는 비밀키 d를 모르고 임의의 평문 M에 대한 서명 $C(C^e=M \bmod 2^l\cdot n)$를 계산하는 것 또한 RSA 서명을 계산하는 것만큼 어렵다는 것을 의미한다. 따라서, Rivest-Shamir-Adleman의 RSA 서명이 안전하다고 한다면 제안된 블럭 보호 방법에 의한 서명 방식 또한 안전하다. 본 논문에서는 제안된 블럭 보호 방법의 위와같은 특성을 이용하여 인증키 분배와 메시지의 비밀성이 보장되는 다중 서명 방식을 제안한다.

V. 블럭 보호 방법의 응용

5.1 인증 키 분배 방식

사용자 A와 B는 DES와 같은 공통의 관용키 암호 시스템을 사용한다고 하고 A가 B에게 큰 사이즈의 메세지를 보내려고 한다고 하자. 이 경우, 사용자 A, B가 공통의 암호 키 K를 서로 공유하고 있다면 K를 관용키 암호시스템의 키로 하여 메시지를 B에게 비밀리에 전달할 수 있을 것이다. 만일 A와 B가 RSA 키만을 가지고 있을 뿐 어떤 공통의 키도 공유하고 있지 않을 경우 우리는 RSA 암호를 이용하여 관용키 암호를 위한 공통 키 K를 분배할 수 있다. 이때, A와 B 사이에 일방향 통신만이 가능하다면 사용자 A는 K에 대한 자신의 서명을 B의 공개키로 암호화시킴으로서 B와 키 K를 안전하게 공유할 수 있다. 이 경우, B는 자신이 수신한 정보를 A가 보냈다는 것을 확인할 수 있으며 A는 암호문으로부터 키 정보를 추출할 수 있는 사용자가 B 밖에 없음을 확신할 수 있다. 이러한 형태의 키 분배 방법을 인증 키 분배(Authenticated Key Distribution)이라고 한다.

RSA를 사용하여 두 사용자가 인증 키를 공유하고자 할 경우 두 사용자가 보유한 RSA 키의 크기 차이로 인한 블럭 보호 문제가 발생한다. 두 사용자간의 인증 키 분배에서 생기는 블럭 보호 문제를 해결하기 위하여 제안된 방법으로는 Kohnfelder의 방법이 가장 잘 알려져 있다[2]. 그러나, Kohnfelder의 방법은 사용자 사이의 RSA 키 크기에 따라 암호화하는 순서를 변경해야 하는 문제점이 있다.

본 절에서는 4절에서 제안된 블럭 보호 방법을 사용하여 사용자의 RSA 키 크기에 관계없이 사용자 상호간에 인증된 키를 분배하는 방법을 제안하고자 한다. n_A, n_B를 각각 사용자 A, B의 RSA modulus라고 하고, (e_A, n_A)는 사용자 A의 공개키 (e_B, n_B)는 사용자 B의 공개키라 하자. A가 B에게 인증 키 $K<n_A$를 전달하기 위하여 다음과 같은 C를 계산하여 보낸다.

$$l=\begin{cases} 0 & \text{if } 2\cdot n_A<n_B \\ 2^{l-1}\cdot n_B<2\cdot n_A<2^l\cdot n_B & \textit{otherwise} \end{cases}$$

$$e_A\cdot d_A=1 \bmod \Phi(n_A),$$
$$C=((2\cdot K+1)^{d_A} \bmod (2\cdot n_A))^{e_B} \bmod (2^l\cdot n_B)$$

암호문 C를 수신한 B는 l을 계산하고 l에 대응되는 비밀키 d를 계산한다.

$$d=\begin{cases} e_B^{-1} \bmod \phi(n_B) & \text{if } l=0, 1 \\ e_B^{-1} \bmod 2^{l-1}\cdot\phi(n_B) & \textit{otherwise} \end{cases}$$

C로 부터 A가 보내 온 비밀 공유키 K를 계산하는 과정은 다음과 같다.

$$2\cdot K+1=(C^d \bmod (2^l\cdot n_B))^{e_A} \bmod (2\cdot n_A)$$

$C_1=(2\cdot M+1)^{d_A} \bmod (2\cdot n_A)$라 하면 암호문 C로 부

터 C_1을 계산할 수 있는 사용자는 비밀키 d를 계산할 수 있는 B 밖에 없으며 키 K로 부터 C_1을 계산할 수 있는 사용자는 A 밖에 없다.

이미 3절에서 설명하였듯이 암호해독자는 암호문 C로 부터 C_1의 하위 l비트를 구할 수 있다. 그러나 C_1의 l비트 정보로부터 키 K의 어떤 부분 정보도 얻을 수 없다. 따라서, 제안된 블럭 보호 방법은 RSA를 이용한 인증 키 분배 방법에 응용 가능하다.

5.2 다중 서명 방식

4절에서 제안된 블럭 보호 방법은 다중 서명 방식에 응용될 수 있다. 본 절에서 제안하는 다중 서명 방식은 Harn과 Kiesler의 다중 서명 방식[6]과 달리 각 서명자들이 하나의 RSA 키만을 사용하여 서명과 서명문의 보호를 동시에 실현할 수 있다. 따라서, 다중 서명된 평문의 내용은 수신자에게 비밀리에 전달된다. 또한, 제안된 방식은 Okamoto의 방식과 달리 메세지 크기가 증가되지는 않으나 서명자의 수에 따라 서명의 길이가 확장된다. 이제 $U_1, U_2, ..., U_k$가 평문 M에 서명하고자 하는 사용자라 하고 $(e_1, n_1), (e_2, n_2), ..., (e_k, n_k)$를 각 사용자들이 갖고 있는 공개키라 하자. 또한 U_{k+1}을 다중 서명의 수신자라 하고 (e_{k+1}, n_{k+1})가 수신자 U_{k+1}의 공개키라 하자. 각 사용자들은 공개키 정보로부터 l_i를 계산할 수 있다.

$$l_i=\begin{cases} 1 & \text{if } i=1 \text{ or } 2^{l_i-1}n_{i-1}<2n_i \\ 2^{l_i-1}n_i<2^{l_i-1}n_{i-1}<2^{l_i}n_i & otherwise \end{cases}$$

각 서명자들이 수신자 U_{k+1}에게 평문 $M<n_1$에 대한 비밀을 유지하면서 서명문을 전달하는 방법은 다음과 같다.

- 서명자 U_1에 의한 서명
  1) $e_1 \cdot d_1 = 1 \bmod \phi(n_1)$이 되는 비밀키 d_1을 계산한다.
  2) $S_1=(2\cdot M+1)^{d_1} \bmod 2\cdot n_1$
  3) $C_1=S_1^{e_1} \bmod 2^{l_2}\cdot n_2$
  4) C_1을 2번째 서명자 U_2에게 전송
- 서명자 U_i에 의한 서명(i = 2, ..., k)
  1) $e_i \cdot d_i = 1 \bmod 2^{l_i-1}-1\cdot\phi(n_i)$인 비밀키 d_i 계산
  2) $S_{i-1}=C_{i-1}^{d_i} \bmod 2^{l_i}\cdot n_i$
  3) S_{i-1}이 평문 M에 대한 서명자 $U_1, U_2, ..., U_{i-1}$의 다중 서명임을 확인
  4) $S_i=S_{i-1}^{d_i} \bmod 2^{l_i}\cdot n_i$
  5) $C_iS_i^{e_{i+1}} \bmod 2^{l_{i+1}}\cdot n_{i+1}$
  6) C_i를 $(i+1)$번째 서명자 U_{i+1}에게 전송
- C_k의 수신자 C_{k+1}에 의한 서명의 검증
  1) $e_{k+1}\cdot d_{k+1}=1 \bmod 2^{l_k+1}-1\cdot\phi(n_{k+1})$인 비밀키 d_{k+1} 계산
  2) $S_k=C_k^{d_{k+1}} \bmod 2^{l_{k+1}}\cdot n_{k+1}$
  3) S_k가 평문 M에 대한 서명자 $U_1, U_2, ..., U_k$의 서명임을 확인한다.

위에서 S_i가 평문 M에 대한 서명자 U_1, U_2, U_i의 서명임을 확인하는 과정은 다음과 같다.

$$S_{j+1}=S_j^{e_j} \bmod 2^{l_j}\cdot n_j(j=i, i-1, ..., 2),\ 2\cdot M+1 = S_1^{e_1} \bmod 2\cdot n_1$$

서명자 U_i가 서명자 U_{i+1}에게 보내는 C_i와 서명자 $U_1, U_2, ..., U_i$의 다중 서명 S_i 사이에는 다음과 같은 관계식이 성립한다.

$$S_i \bmod 2^{l_{i+1}}=C_i^{d'} \bmod 2^{l_{i+1}}(e_{i+1}\cdot d'=1 \bmod 2^{l_{i+1}-1})$$
$$S_i \bmod 2^{l_i}=C_i^{d''} \bmod 2^{l_{i+1}}(e_i\cdot d''=1 \bmod 2^{l_i-1})$$

서명자 U_{i+1}의 비밀키를 모른다 하더라도 C_i로 부터 서명 S_i의 최하위 l_{l+1} 비트를 구할 수 있으나 S_i의 상위 $|n_{i+1}|$ 비트는 구할 수 없다. 또한, 서명자 U_i의 비밀키를 모르고서도

서명 S_{i-1}로 부터 서명 S_i의 최하위 l_i 비트를 계산할 수 있으나 S_i의 상위 $|n_i|$ 비트를 구할 수 없다. 제안된 다중 서명의 안전성은 RSA 키 $n_1, n_2, ..., n_k, n_{k+1}$중에서 가장 작은 RSA 키의 안전성에 의존하게 된다. 따라서, $n_1, n_2, ..., n_{k+1}$는 모두 같은 비트 사이즈를 갖는것이 좋다.

제안된 방법에 의하여 확장되는 서명문의 비트 길이는 얼마나 되는가? $L=\max(|n_1|, |n_2|, ..., |n_k|)$라 하면 제안된 방법에 의하여 생성된 최종 서명문 S_k의 비트 길이는 $|S_k|<L+k+1$이다. 따라서, 확장되는 비트 길이는 서명자의 수 k 비트 보다 작거나 같으며, Okamoto의 다중 서명 방법과 달리 각 서명자들이 갖

는 RSA 키의 크기와는 무관하다. 예를들어, 6명의 사용자가 다음과 같은 비트 길이를 갖는 RSA 키를 사용한다고 하자.

$|n_1| = |n_3| = |n_5| = 768$, $|n_2| = |n_4| = |n_6| = 512$

Okamoto 방법에서는 총 771 비트가 증가되었으나 본 다중 서명 방법은 최대 6비트만이 증가될 뿐이다. 따라서, 각 서명자들이 서로 다른 크기를 갖는 RSA 키를 사용할 경우 매우 효율적이다. 만일 각 서명자의 RSA 키 $n_1, n_2, ..., n_k$가 모든 같은 비트 길이를 갖는다면 확장되는 비트 길이는 최대 k 비트로서 Okamoto의 방법과 동일하다. 〈표 1〉에서는 Okamoto 방식, Harn-Kiesler 방식과 제안된 방식을 비교하였다.

표 1. 다중서명 방식 비교

서명 방식	*Okamoto*	*Harn-Kiesler*	제안된 방식
서명순서	제약없음	제약있음	제약없음
평문의 비밀성	제공안됨	제공됨	제공됨
확장되는 비트수	$\geq k$	없음	$\leq k$

(k는 다중서명에 참가한 서명자 수)

5.3 다중 서명의 간단한 예

본 절에서는 간단한 예제를 통하여 제안된 다중 서명 방식의 서명 과정과 검증 과정을 소개하고자 한다. 다중 서명에 참여하는 3명의 서명자와 최종 서명 수신자가 사용하는 RSA 키들은 다음과 같다(모든 데이타들은 16진수로 표기).

- 서명자 1

  $n_1 = 3ac01517fd1e6efab$, $\phi(n_1) = 3ac0151703eaf390$, $e_1 = 9b$
- 서명자 2

  $n_2 = 87c5b9a19aa73e8d$, $\phi(n_2) = 87c5b9a0149f91dc$, $e_2 = af$
- 서명자 3

  $n_3 = 201c2081229c57dd$, $\phi(n_3) = 201c20804a9854dc$, $e_3 = b5$
- 서명 수신자

  $n_4 = bf09bacc431bed19$, $\phi(n_4) = bf09baca858ca6b4$, $e_4 = fb$

서명자 1, 2, 3 차례로 서명할 경우 공개키 n_1, n_2, n_3로부터 l_1, l_2, l_3, l_4를 계산할 수 있다.

$l_1 = 1$, $l_2 = 1$, $l_3 = 4$, $l_4 = 2$

평문 $M = 2d4d8f$의 다중 서명 과정은 다음과 같다.

- 서명자 1에 의한 서명 과정
  - 비밀키 d_1 계산:

    $d_1 = e_1^{-1} \bmod \Phi(n_1) = 1d908ebdf4c23533$
  - 평문 M의 서명문 S_1 계산:

    $S_1 = (2 \cdot M + 1)^{d_1} \bmod 2 \cdot n_1 = 419873fcee5dfc5f$
  - 암호문 C_1 계산:

    $C_1 = (S_1)^{e_1} \bmod 2^{l_1} \cdot n_2 = 26f6a128c16aa729$
  - C_1을 서명자 2에게 전송
- 서명자 2에 의한 서명 과정
  - 비밀키 d_2 계산:

    $d_2 = e_2^{-1} \bmod 2^{l_2 - 1} \cdot \phi(n_2) = 35887976b4bfa5c3$
  - 서명자 1의 서명문 S_1 복호화:

    $S_1 = (C_1)^{d_2} \bmod 2^{l_2} \cdot n_2 = 419873fcee5dfc5f$
  - 서명자 2의 서명문 S_2 계산:

    $S_2 = (S_1)^{d_2} \bmod 2^{l_2} \cdot n_2 = 3de593c4d2695f83$
  - 암호문 C_2 계산:

    $C_2 = (S_2)^{e_2} \bmod 2^{l_3} \cdot n_3 = 62c88017dbba5de3$
  - C_2를 서명자 3에게 전송
- 서명자 3에 의한 서명 과정
  - 비밀키 d_3 계산:

    $d_3 = e_3^{-1} \bmod 2^{l_3 - 1} \cdot \phi(n_3) = 375980281abfdd3d$
  - 서명자 2의 서명문 S_2 복호화:

    $S_2 = (C_2)^{d_3} \bmod 2^{l_3} \cdot n_3 = 3de593c4d2695f83$
  - 서명자 3의 서명문 S_3 계산:

    $S_3 = (S_2)^{d_3} \bmod 2^{l_3} \cdot n_3 = 842097ed0b7c0a63$
  - 암호문 C_3 계산:

    $C_3 = (S_3)^{e_4} \bmod 2^{l_4} \cdot n_4 = 1907a653371482c4f$

- C_3를 서명 수신자에게 전송

평문 M에 대한 다중 서명 S_3의 암호문 C_3를 수신한 서명 수신자 4는 다음과 같은 과정에 의하여 다중 서명을 검증한다.

- 비밀키 d_4 계산:

$$d_4 = e_4^{-1} \bmod 2^{l_4-1} \cdot \phi(n_4) = c5e352514600dda3$$

- C_3로 부터 다중 서명 S_3 복호화:

$$S_3 = (C_3)^{d_4} \bmod 2^{l_4} \cdot n_4 = 842097ed0b7c0a63$$

- 서명자 1, 2, 3의 공개키 e_1, e_2, e_3에 의하여 S_0를 계산한다.

$$S_2 = (S_3)^{e_3} \bmod 2^{l_3} \cdot n_3 = 3de593c4d2695f83$$

$$S_1 = (S_2)^{e_2} \bmod 2^{l_2} \cdot n_2 = 419873fcee5dfc5f$$

$$S_0 = (S_1)^{e_1} \bmod 2^{l_1} \cdot n_1 = 5a9b1f$$

- 만일 $S_0 = 2 \cdot M + 1$이면 S_3는 평문 M에 대한 다중 서명으로 받아들인다.

Ⅵ. 결 론

본 논문에서는 입력되는 데이터의 크기에 따라 암호 블럭의 크기를 가변시킬 수 있는 새로운 RSA 블럭 보호 방법을 제안하였다. 제안된 방법은 인증 키 분배와 서명 순서에 제한을 받지 않는 다중 서명 방식에 적용 가능하며, 각 서명자는 하나의 RSA 키만으로 서명된 메시지를 수신자에게 비밀리에 전달 할 수 있다. 또한, 제안된 방식은 서명문의 크기가 서명자의 수에 따라 증가된다. 이때, 확장되는 비트 사이즈는 서명자의 수 보다 작거나 같으며 각 서명자가 갖는 RSA 키의 크기와는 무관하다. 따라서, 임의의 크기를 갖는 사용자 그룹에서 서명 순서에 제한을 받지 않는 다중 서명에 가장 적합한 방식이라고 생각된다.

참 고 문 헌

1. R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystem", *Commun. ACM*, 1978, 21, (2), pp. 120-126.
2. L. M. Kohnfelder, "On the signature reblocking problem in public-key cryptography", *Commun. ACM*, 1978, 21, (2) pp. 179.
3. J. Levine and J. V. Brawley, "Some cryptographic applications of permutation polynomials", *Cryptologia*, 1977, 1, pp. 76-92.
4. K. Itakura and K. nakamura, "A public-key cryptosystem suitable for digital multisignatures", *NEC Research and Develop.*, 1983, 71, pp. 1-8.
5. T. Okamoto, "A digital multisignature scheme using bijective public-key cryptosystems", *ACM Trans. Computer Systems*, 1988, 6, (8), pp. 432-441.
6. L. Harn and T. Kiesler, "New scheme for digital signatures", *Electron. Lett.*, 1989, 25, (22), pp. 1527-1528.

박 상 준(Sangjoon Park) 정회원

1960년 6월 25일생
1984년 2월 : 한양대학교 수학과 졸업(이학사)
1986년 2월 : 한양대학교 대학원 수학과 졸업(이학석사)
1986년 1월~현재 : 한국전자통신연구소 선임연구원
1995년 3월~현재 : 성균관대학교 대학원 정보공학과 박사과정

원 동 호(Dong Ho Won) 정회원

1949년 9월 23일생
1976년 2월 : 성균관대학교 전자공학과 졸업(공학사)
1978년 2월 : 성균관대학교 대학원 전자공학과 졸업(공학석사)
1988년 2월 : 성균관대학교 대학원 전자공학과 졸업(공학박사)
1978년 4월~1980년 3월 : 한국전자통신연구소 연구원
1985년 9월~1986년 8월 : 일본 동경공대 객원연구원
1982년 3월~현재 : 성균관대학교 공과대학 정보공학과 교수
1991년~현재 : 한국통신정보보호학회 편집이사
1996년 4월~현재 : 정보화추진위원회 자문위원
※주관심분야 : 암호이론, 정보이론