通信情報保護學會論文誌
第 7 卷 第 2 號. 1997. 6

효율적인 RSA 다중 서명 방식

박 상 준*, 박 상 우*, 원 동 호**

Efficient RSA Multisignature Scheme

Sangjoon Park*, Sangwoo Park*, Dong-Ho Woo**

요 약

본 논문에서는 서명의 순서에 제한을 받지 않으며 비트 확장을 발생시키지 않는 RSA 다중 서명 방식을 제안하고자 한다. 제안된 방법에서는 모든 사용자들이 동일한 비트 길이를 갖고 상위 l 비트 패턴이 같은 RSA modulus를 사용한다. 이러한 형태의 RSA 키들은 Levine과 Brawley가 제안한 반복 지수승 연산 기법(repeated exponentiation)과 함께 다중 서명 방식에 응용된다. 본 논문에서 제안된 다중 서명 방식은 Levine과 Brawley의 반복 지수승 연산을 이용한 Kiesler-Harn의 방식보다 다중 서명 생성에 요구되는 계산량을 줄일 수 있다. m명의 사용자가 다중 서명에 참여할 경우 Kiesler-Harn 방식은 평균 $1.5m$회의 지수승 연산이 요구되나 제안된 방식에서는 $(1 + \frac{1}{2^{l-1}})$ m회의 지수승 연산이 요구된다. 따라서, l이 충분히 클 경우($l \geq 32$) 다중 서명에 필요한 지수승 연산의 수는 거의 m과 같게 된다.

Abstract

In this paper, we propose an RSA multisignature scheme with no bit expansion in which the signing order is not restricted. In this scheme we use RSA moduli with the same bit length, the most l bits of which are same. The proposed scheme is based on these RSA moduli and a repeated exponentiation of Levine and Brawley. Kiesler and Harn first utilize the repeated exponentiation technique in their multisignature scheme, which requires $1.5m$ exponentiations for signing, where m is the number of signers. However, the proposed scheme requires $(1 + \frac{1}{2^{l-1}})$ m exponentiation. So if l is sufficiently large ($l \geq 32$), then we can neglect the vaue $\frac{1}{2^{l-1}}$

. 한국전자통신연구원
.. 성균관대학교 정보공학과

1. 서 론

RSA 암호는 인수 분해 문제의 어려움에 근거한 공개키 암호시스템으로 데이터 암호 및 디지털 서명에 적용 가능하다[1]. 그러나, 두명의 사용자간에 인증 키를 분배하고자 하거나 여러 명이 하나의 서류에 다중 서명(multisignature) 하고자 할 경우에 각 사용자들이 소유하고 있는 RSA modulus 값들의 차이로 인하여 블럭 보호 문제(blocking problem)를 유발시킨다.

RSA 암호를 이용한 다중 서명 방식은 서명문에서 비트 확장을 발생시키는 방식과 비트 확장을 발생시키지 않는 방식 두 가지로 분류할 수 있다. 비트 확장을 발생시키는 대표적인 다중 서명 방식은 Okamoto 다중 서명 방식이 있다[2]. Okamoto 방식은 RSA와 같이 전단사 함수로 구성되는 공개키 암호에 적용될 수 있다. 서명 순서에 제한이 없으나 RSA 키의 비트 길이와 서명자의 수에 따라 메시지의 길이가 확장된다.

비트 확장을 발생시키지 않는 다중 서명 방식은 Harn과 Kiesler가 제안한 2가지 방식이 있다. 하나는 Kohnfelder 방법을 일반화시킨 것으로 서명에 참여한 서명자들이 소유한 RSA modulus중 가장 크기가 작은 RSA modulus 값을 갖은 사용자부터 차례로 서명을 한다. 그러므로, RSA 키값에 따라 서명의 순서가 제약되며 다중 서명중에 새로운 사용자가 서명에 참여할 수 없는 단점이 있다[3][4]. 또 다른 방법은 RSA original paper에 소개된 Levine과 Brawley의 방법을 일반화시킨 방법이다. 모든 사용자는 k 비트 길이를 갖는 RSA modulus를 갖으며 다중 서명에 참여하는 각 서명자는 2^{k-1} 보다 작은 서명이 생성될 때까지 자신의 비밀키에 의한 지수승 연산을 반복적으로(repeated exponentiation) 수행한다[5][6]. 따라서, 중간 서명문과 최종 서명문의 비트 길이는 k-1 비트로 비트 확장을 발생시키지 않으며, 서명 순서에 제한이 없고 중간에 새로운 사용자가 서명에 참여할 수도 있다. 그러나, 각 서명자들은 RSA 서명문의 비트 길이가 k-1 비트가 될 때까지 반복적으로 지수승 연산을 수행하여야 하므로 계산량이 증가되는 문제점이 있다. 일반적으로, m명의 사용자가 서명에 참여하고자 할 경우 최종 서명문이 만들어 질 때까지 수행하여야 하는 지수승 연산 수는 평균 1.5 m 회이다.

본 논문에서는 서명 순서에 제한을 받지 않고 비트 확장을 발생시키지 않는 새로운 RSA 다중 서명 방식을 제안하고자 한다. 제안된 방법은 Kiesler와 Harn의 방법을 일반화시켰다. Kiesler-Harn의 서명 방식에서 시스템 내의 모든 사용자는 모두 비트 길이가 동일한 RSA 키를 사용한다. 이 경우, RSA modulus의 최상위 비트 모두 '1'로서 동일하게 된다. 본 논문에서 제안되는 다중 서명 방식은 시스템 내의 모든 사용자가 동일한 비트 길이를 갖고 상위 1 비트가 동일한 비트 패턴을 갖는 RSA modulus를 사용함으로서 평균 $(1 + \frac{1}{2^{l-1}})$ m회의 지수승 연산만으로다중 서명을 생성한다. 그러므로, $l \geq 32$인 경우 각 서명자들이 1회 이상의 지수승 연산만을 수행할 확률은 거의 0에 가까우므로 Kiesler와 Harn의 방식 보다 효율적이다.

본 논문은 모두 6개 절로 구성된다. 2절에서는 Levine과 Brawley의 반복 지수승 연산을 이용한 Kiesler와 Harn의 다중 서명 기법을 소개하고 3절에서는 동일한 비트 길이를 갖고 상위 l 비트 패턴이 같은 RSA 키들 사이의 블럭 보호 문제(blocking problem)을 기술하였다. 4절은 3절에서 소개한 RSA 키를 사용한 다중 서명 방식을 소개하고 Kiesler-Harn 방식과 계산량을 비교.분석하였다. 5절에서는 동일한 비트 길이를 갖고 상위 l 비트 패턴이 같은 RSA 합성수 $n = pq$을 구성하기 위하여 strong RSA 소수 p, q를 생성하는 방법을 제안하였다. 6절

은 결론부이다.

2. Kiesler-Harn의 다중 서명 기법

본 절에서는 Kiesler-Harn의 다중 서명 기법을 소개하고자 한다. Kiesler-Harn 방식은 RSA 블럭 보호 문제를 해결하기 위하여 Levine과 Brawley가 제안한 반복 지수승 연산(repeated exponentiation)을 다중 서명 방식에 적용한 것이다[5][6]. 모든 시스템 내의 사용자들은 k 비트 길이를 갖는 RSA modulus를 갖는다. 사용자 U_i의 RSA modulus를 n_i라하고, (e_i, n_i)를 U_i의 공개키, (d_i, n_i)를 사용자 U_i의 비밀키라고하자. U_1, U_2, · · · , U_m를 서명자라 하면 서명 생성 과정은 다음과 같다.

- 사용자 U_1의 서명

  U_1은 서명하고자 하는 메시지 $0 \le M < 2^{k-1}$을 자신의 비밀키를 사용하여 $S_1 = M^{d_1} \bmod n_1$을 계산하고, 만일 $S_1 \ge 2^{k-1}$ 이면 $S_1 < 2^{k-1}$이 될 때까지 $S_1 = S_1^{d_1} \bmod n_1$ 계산 과정을 반복적으로 수행한다. $S_1 < 2^{k-1}$을 다음 서명자 U_2에게 전달한다.

- 사용자 U_i에 의한 서명 $(i=2, 3, \cdots, m)$

  U_{i-1}으로 부터 수신된 서명 S_{i-1}에 대하여 자신의 비밀키를 사용하여 $S_i = S_{i-1}^{d_i} \bmod n_1$을 계산하고, 만일 $S_i \ge 2^{k-1}$이면 $S_i < 2^{k-1}$이 만족될 때까지 $S_i = S_i^{d_i} \bmod n_1$ 계산 과정을 반복적으로 수행한다. $S_i < 2^{k-1}$을 다음 서명자 U_{i+1}에게 전달한다.

서명 수신자는 서명 S_m이 메시지 M에 대한 서명자 $U_1, U_2, \cdots, U_m$의 다중 서명임을 다음과 같은 과정으로 확인한다.

- $j = k, k-1, \cdots, 1$에 대하여, $S_{j-1} = S_j^{e_j} \bmod n_j$를 계산하고 $S_{j-1} \ge 2^{k-1}$이면 $S_{j-1} < 2^{k-1}$이 만족될 때까지 반복하여 $S_{j-1} = S_{j-1}^{e_j} \bmod n_j$을 계산한다. 마지막으로, S_0가 메시지 M과 같은지 확인한다.

각 사용자들의 RSA 키 n_i는 k 비트 길이를 가지므로 임의의 $0 \le x < n_i$가 $0 \le x < 2^{k-1}$이 될 확률은 $\frac{1}{2}$ 보다 크다.

$$Pr[0 \le x < 2^{k-1} \mid 0 \le x < n_i] = \frac{2^{k-1}}{n_i} > \frac{2^{k-1}}{2^k} = \frac{1}{2}$$

각 서명자들이 2^{k-1} 보다 작은 값을 갖는 서명을 생성하기 위하여 반복적으로 수행하는 지수승의 횟수는 평균 1.5회 보다 작다. 따라서, 다중 서명 S_m 을 생성 (혹은 검증)하기 위한 지수승 연산은 평균 $1.5m$이 된다.

3. 블럭 보호 방법

본 절에서는 Kiesler-Harn의 방법을 보다 확장시켜 시스템 내의 모든 사용자들이 k비트 RSA modulus를 사용하며 각 RSA modulus의 최상위 l 비트가 동일한 경우의 RSA 블럭 보호 방법을 검토하고자 한다. RSA modulus n이 k 비트이고 최상위 l 비트 값이 C가 된다고 하면 n은 다음과 같이 나타낼 수 있다.

$$n = C \cdot 2^{k-1} + B \quad (0 \le B < 2^{k-l})$$

이 경우 $0 \le x < n$인 임의의 랜덤수 x가 $0 \le x < C \cdot 2^{k-1}$이 될 확률은 $1 - 2^{-l+1}$ 보다 크다.

$$Pr[0 \le x < C \cdot 2^{k-1} \mid 0 \le x < n] = \frac{C \cdot 2^{k-1}}{n} = 1 - \frac{B}{n} > 1 - \frac{2^{k-1}}{2^{k-1}} = 1 - 2^{-l+1}$$

RSA 지수승 연산은 pseudo-random permutation으로 생각할 수 있으며, 따라서, 평

문 M에 관계없이 RSA 암호문 $C = M^e \bmod n$는 0와 n 사이의 랜덤한 수로 변환된다. 본 절에서 제안된 형태의 RSA 키는 Levine과 Brawley의 반복 지수승 연산 기법과 결합하여 사용할 경우 인증 키 분배, 다중 서명 등에 적용시킬 수 있는 효과적인 RSA 블럭 보호 방법으로 사용될 수 있다.

4. 효율적인 다중 서명 기법

$U_1, U_2, \cdots, U_m$을 다중 서명에 참여하는 서명자들이라 하고 $n_1, n_2, \cdots, n_m$는 각 서명자들의 RSA 키라 하자. 서명자들의 RSA 키들은 모두 k 비트 길이를 갖고 최상위 l 비트 패턴이 같다고 하자.

$$n_i = C \cdot 2^{k-l} + B_i \ (0 < B_i < 2^{k-l}, 2^{l-1} < C < 2^l)$$

e_i, n_i는 서명자 U_i의 RSA 공개키이고 d_i, n_i는 서명자 U_i의 RSA 비밀키이다. 이 경우 서명자 $U_1, U_2, \cdots, U_m$에 의하여 평문 $M < C \cdot 2^{k-l}$에 대한 다중 서명을 생성하는 과정은 다음과 같다.

- 1번째 서명자 U_1
  - ① $S_1 = M^{d_1} \bmod n_1$
  - ② 만일 $S_1 \geq C \cdot 2^{k-l}$이면 $S_1 < C \cdot 2^{k-l}$이 만족될 때까지 $S_1 = S_1^{d_1} \bmod n_1$을 반복적으로 계산한다.
  - ③ S_1을 서명자 U_2에게 전송한다.
- i번째 서명자 ($i = 2, 3, \cdots, m-1$)
  - ① $S_i = S_{i-1}^{d_i} \bmod n_i$
  - ② 만일 $S_i \geq C \cdot 2^{k-l}$이면 $S_i < C \cdot 2^{k-l}$이 만족될 때까지 $S_i = S_i^{d_i} \bmod n_i$을 반복적으로 계산한다.
  - ③ S_i를 서명자 U_{i+1}에게 전송한다.
- m 번째 서명자
  - ① $S_m = S_{m-1}^{d_m} \bmod n_m$
  - ② S_m를 서명 수신자에게 전송한다.

서명 수신자는 S_m이 서명자 $U_1, U_2, \cdots, U_m$의 다중 서명이 됨을 검증하기 위하여 $j = m, m-1, \cdots, 1$에 대하여 차례로 다음의 과정을 수행한 후 마지막 S_0가 메시지 M이 되는지 확인한다.

- ① $S_{j-1} = S_j^{e_j} \bmod ni$ 계산한다.
- ② 만일 $S_{j-1} \geq C \cdot 2^{k-l}$이면 $S_{j-1} < C \cdot 2^{k-l}$이 만족될 때까지 $S_{j-1} = S_{j-1}^{e_j} \bmod ni$를 반복적으로 수행한다.

3절에서 기술하였듯이 $S_{i-1}^{d_i} \bmod n_i < C \cdot 2^{k-l}$ 이 성립할 확률은 $1 - 2^{-l+1}$ 보다 크다. 따라서, $l = 32$ (or 48, 64)인 경우 각 서명자가 반복해서 자신의 비밀키로 지수승 연산을 수행하게 될 확률은 거의 없다. 그러므로, 서명 S_m이 생성되기까지 각 서명자들이 수행한 지수승 연산의 총 횟수는 평균 $(1 + 2^{-l+1}) \cdot m$ 보다 작다. 마찬가지로, 최종 S_m를 수신한 수신자는 평균 $(1 + \frac{1}{2^{l-1}}) \cdot m$ 회의 지수승 연산만으로 검증 가능하다. 제안된 방법에서는 각 서명자들이 현실적으로 단 1회의 지수승 연산만을 수행하기 때문에 Kiesler와 Harn이 제안한 방법 보다[5] 훨씬 효과적이다. 예를 들어 $l = 32$이고 $m = 20$이라 하면 Kiesler-Harn 방법에서는 평균 $1.5 \times 20 = 30$회의 지수승 연산이 요구되나 제안된 방법에서는 평균 $(1 + 2^{-31}) \times 20$이 된다. 이 경우 2^{-31}은 무시할 수 있으므로 20회 정도의 지수승 연산을 수행한다고 할 수 있다.

5. 키 생성 방법

본 절에서는 3, 4절에서 제안된 형태의 RSA 키를 생성하는 방법에 대하여 기술하고자 한다. 먼저 키관리 센터에서는 시스템에서 사용하고자 하는 RSA 키 비트 길이 k와 l 비트 패턴 C를 랜덤하게 생성하여 모든 사용자들에게 공개한다. k는 편의성을 위하여 짝수라고 하자. 이때 각 사용자들의 RSA 키 n은 비트 길이가 k이고 최상위 l 비트 패턴이 C가 되어야 할 뿐 아니라, RSA 암호의 안전성을 위하여 n은 2개의 strong prime의 곱이 되어야 한다. 다음은 본 논문에서 제안된 형태의 strong RSA 키를 생성하는 과정이다[7].

[step 1.] 사용자 U_i는 k-l 비트 랜덤수 R를 생성하여 $N = C \cdot 2^{k-l} + R$이라 놓는다.

[step 2.] $\frac{k}{2}$ 비트 랜덤수 P를 생성한다.

[step 3.] $\frac{k}{2}$ -l-t비트 소수 p_i', q_i'을 생성한다.

[step 4.] $s = \frac{P}{2 \cdot p_i'}$ 을 계산

[step 5.] $p_i = 2 \cdot p_i' \cdot s + 1$이 소수가 아니면 $s = s + 1$하고 step 5.를 다시 시행한다.

[step 6.] $Q = \left\lfloor \frac{N}{p_i} \right\rfloor$를 계산한다.

[step 7.] $s = \frac{Q}{2 \cdot q_i'}$ 을 계산

[step 8.] $q_i = 2 \cdot q_i' \cdot s + 1$이 소수가 아니면 $s = s + 1$하고 step 8.를 다시 시행한다.

[step 9.] $n_i = p_i \cdot q_i$를 계산하고 $\left\lfloor \frac{n_i}{2^{k-l}} \right\rfloor = C$이 성립하지 않으면 step 1. 으로 돌아간다.

위와 같은 과정을 통해서 생성된 n_i의 상위 l 비트 패턴이 C로 모두 동일하지만 n_i의 소인수 p_i와 q_i들은 랜덤한 비트 패턴을 갖는다. step 3.에서 t 값은 step 5.와 step 8.에서 증가되는 s값에 의하여 합성수 n_i의 최상위 l 비트 패턴 C가 변화되지 않도록 하기 위하여 설정한 변수이다. 현실적으로 $t = 16$으로 하면 step 9.에서 생성되는 RSA modulus n_i의 상위 l 비트는 거의 대부분의 경우 C가 된다. step 3.에서 생성된 p_i', q_i'은 $p_i - 1$과 $q_i - 1$의 $\frac{k}{2}$ -l-t비트 소인수이므로 소수 p_i, q_i는 strong prime이 된다. 또한, step 3.에서 생성된 소수 p_i', q_i'은 $p_i' - 1$, $q_i' - 1$이 큰소인수를 갖도록 할 수 있다[8][9]. 위와 같은 방법으로 생성된 소수 p_i, q_i와 합성수 n_i는 $k = 768$, $l = 32$, $t = 16$일 경우 다음과 같은 특징을 갖는다.

- 사용자의 모든 RSA modulus는 768 비트이며 상위 32 비트는 C와 같다.
  $2^{767} < n_i < 2^{768}, C = \left\lfloor \frac{n_i}{2^{736}} \right\rfloor$
- 소수 p_i, q_i는 384 비트이고 $p_i - 1$, $q_i - 1$은 각각 336 비트 길이를 갖는 소인수 p_i', q_i'을 갖는다.
- 다중 서명 방식에서 서명 생성 및 검증시에 수행되는 지수승 연산의 횟수는 평균 $(1 + 2^{-31})m$이다.

6. 결 론

본고에서는 시스템 내의 모든 사용자들이 동일한 비트 길이를 갖고 상위 l 비트 패턴이 동일한 RSA modulus를 사용할 경우 RSA에 근거한 다중 서명에서 블럭 보호 문제가 거의 발생하지 않으며 블럭 보호 문제가 발생할 확률은 $\frac{1}{2^{l-1}}$ 보다 작음을 보였다. 블럭 보호 문제가 발생할 경우 Kiesler-Harn 다중 서명에서와 같이 Levine과 Brawley의 반복 지수승 연산 기법으

로 해결할 수 있다. 또한, 제안된 방식에서 요구되는 형태의 strong RSA 키를 생성하는 방법을 제안하였다. 본 논문에서 제안된 다중 서명은 다음과 같은 특징을 갖는다.

① 비트 확장을 유발시키지 않는다.
② 서명 순서에 제한을 받지 않는다.
③ 각 서명자가 2회 이상의 지수승 연산을 수행할 확률은 거의 없다.
④ 하나의 RSA 키만을 사용하여 서명을 수신자에게 비밀리에 전송할 수 있다.

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] T. Okamoto, "A digital multisignature scheme using bijective public-key cryptosystems", *ACM Trans. Computer Systems*, 1988, 6, (8), pp.432-441

[3] L. M. Kohnfelder, "On the signature reblocking problem in public-key cryptography", *Commun. ACM*, 1978, 21, (2) pp.179

[4] L. Harn and T. Kiesler, "New scheme for digital signatures", *Electron. Lett.*, 1989, 25, (22), pp.1527-1528

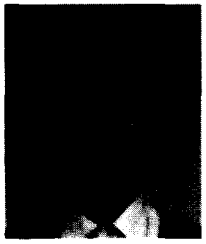
[5] T. Kiesler and L. Harn, "RSA blocking and multisignature schemes with no bit expansion", *Electron. Lett.*, 1990, 26, (18), pp.1490-1491

[6] J. Levine and J. V. Brawley, "Some cryptographic applications of permutation polynomials", *Cryptologia*, 1977, 1, pp. 76-92

[7] L. R. Rivest, "Remarks on a proposed cryptanalytic attack on the M.I.T. public-key cryptosystem", *Cryptologia*, 1978, Vol.2, No. 1,pp. 62-65

[8] G. J. Simmons and M. J. Norris, "Preliminary comments on the M.I.T. public-key cryptosystem", *Cryptologia*, 1977, Vol.1, No.4, pp. 404-414

□ 著者紹介



박 상 준(朴 商 埈, Sangjoon Park)

1984년 2월 한양대학교 자연과학대학 수학과(이학사)
1986년 2월 한양대학교 대학원 수학과(이학석사)
1986년 1월 ~ 현재 한국전자통신연구원 선임연구원

박 상 우

1985년 ~ 1989년 고려대학교 사범대학 수학교육학과(이학사)
1989년 ~ 1991년 고려대학교 대학원 수학과(이학석사 : 응용수학 및 확률론)
1991년 ~ 현재 한국전자통신연구원 연구원

원 동 호(元 東 豪, Dong Ho Won)

1976년 2월 성균관대학교 전자공학과 졸업(공학사)
1978년 2월 성균관대학교 대학원 졸업(공학석사)
1988년 2월 성균관대학교 대학원 전자공학과(공학박사)
1978년 4월 ~ 1980년 3월 한국전자통신연구소 연구원
1985년 9월 ~ 1986년 8월 일본 동경 특공대 객원 연구원
1982년 3월 ~ 현재 성균관대학교 공과대학 정보공학과 교수
1991년 ~ 현재 한국통신정보보호학회 편집이사
1996년 4월 ~ 현재 정보화추진위원회 자문위원

※ 주관심분야 : 암호이론, 정보이론