메시지 분할을 이용한 RSA 다중서명 방식

장종성*, 정광식*, 손진곤*
*한국방송통신대학교 정보과학과
e-mail : hiena@knou.ac.kr

A RSA Multisignature Scheme using Message Partition

Jong Sung Jang*, Kwang Sik Chung*, Jin Gon Shon*
*Dept. of Computer Science,
Graduate School, Korea National Open University

요 약

다중서명에서는 보안적 요소뿐만 아니라 다중서명의 길이, 다중서명의 생성 및 검증 시간, 복호화 시간, 통신량 등도 중요한 요소이다. 기존의 RSA를 그대로 이용한 RSA 원형 다중서명 방식과 RSA를 변형하여 적용한 RSA 변형 다중서명 방식은 다중서명의 생성 및 검증 시간이 각각의 개인서명을 생성하고 검증하는 시간의 합과 비례하여 증가한다는 문제점이 있다. 본 논문은 서명자의 수만큼 분할된 각각의 부분메시지에 대해 RSA 방식을 이용하여 제각기 서명하는 다중서명 방식을 제안한다. 제안된 메시지 분할을 이용한 RSA 다중서명 방식은 블록의 수가 증가하지 않는 경우 기존 방식에 비해 다중서명의 길이가 증가하지 않으면서, 다중서명의 생성 및 검증 시간이 각각 $\frac{1}{L}$로 감소하고, 복호화 시간과 통신량은 각각 약 $\frac{1}{2}$~$\frac{1}{L}$로 감소한다.

1. 서론

보편화된 정보화 환경에서 종이문서가 전자문서로 변화함에 따라 하나의 문서에 여러 명의 서명자가 서명하는 다중서명의 필요성이 증가하였다. 이렇게 컴퓨터 네트워크를 통해 처리되는 전자문서에 대한 다중서명에서 인증, 무결성, 부인방지 등과 같은 보안적 요소뿐만 아니라 다중서명의 길이, 다중서명의 생성 및 검증 시간, 복호화 시간, 통신량 등도 고려되어야 할 중요한 요소이다.

다중서명에 적용될 수 있는 디지털 서명방식은 여러 종류가 있다. 그 중 RSA(Rivest-Shamir-Adleman)[1]는 알고리즘이 단순하고 보다 적은 파라미터를 요구하기 때문에 널리 사용되고 있다[2]. 그러나 RSA를 그대로 적용하여 전체메시지의 동일한 복사본들에 각각 서명하는 RSA 원형 다중서명 방식[3]과 변형된 RSA를 적용하여 하나의 동일한 전체메시지에 모든 서명자가 반복하여 서명하는 RSA 변형 다중서명 방식[2][4][5]은 다중서명의 생성 및 검증 시간이 각각의 개인서명을 생성 및 검증하는 시간의 합과 비례하여 증가하는 문제점이 있다.

본 논문에서는 전체메시지를 서명자의 수만큼 분할하고, RSA 방식을 이용하여 분할된 각각의 부분메시지에 여러 명의 서명자가 제각기 서명하는 다중서명 방식을 제안하고자 한다.

2. 전체메시지 분할 시 시스템 가정

본 논문에서 제안된 메시지 분할을 이용한 RSA 다중서명 방식은 첫 번째 서명자가 서명을 하기에 앞서 전체메시지를 부분메시지들로 분할한다. 전체메시지를 분할함에 있어서는 다음과 같은 사항을 가정한다.

첫째, 전체메시지는 서명자의 수만큼 부분메시지로 분할되고, 각 서명자는 각자에게 할당된 부분메시지에 서명한다.

둘째, 각 부분메시지는 내용적으로 어떠한 의미도 갖지 않는다. 부분메시지가 내용적으로 어떤 의미를 내포한다고 할 경우 차후 어떤 서명자는 전체메시지에 대한 책임을 회피할 수 있다.

셋째, 본 논문에서 제안된 전체메시지 분할 알고리즘을 통해 전체메시지를 균등하게 부분메시지로 분할한다면, 그 부분메시지들은 전체메시지에 대한 대표성을 갖는다. 부분메시지는 전체메시지를 분할한 것이기 때문에 전체메시지와 내용적으로 동일할

수 없지만, 각 서명자가 서명한 부분메시지는 전체 메시지에 대한 대표성을 갖고 있기 때문에 그 서명은 유효하다.

3. 제안방식

3.1. 메시지 분할 및 통합 방법

메시지 분할 방법은 전체메시지 M을 $\{b_1 \| b_2 \| \cdots \| b_I\}$과 같이 비트열로 간주하고, 각 비트열의 인덱스를 서명자 수 L로 법(mod) 연산을 하여 나머지가 같은 비트끼리 부분메시지를 구성한다. 즉 부분메시지 MP_l은 $i \bmod L (1 \le i \le I,\ 1 \le l \le L)$과 같은 연산에 의해 구성된다. 전체메시지의 분할에 대한 알고리즘은 <그림 1>과 같다.

```
Message_Partition(M, L){
  /* M={b1 ‖ b2 ‖ … ‖ bI}은 전체메시지,
     I는 전체메시지의 크기
     L은 서명자 수 */

  for(i=1; i<=I; i++)
    switch(i mod L){
      case 1: MP1={MP1 ‖ bi};
      /* MPl은 부분메시지 */
      case 2: MP2={MP2 ‖ bi};
      ……
      case 0: MPL={MPL ‖ bi};
      default: error Msg;
    }
  if (부분메시지들의 크기가 다른가)
    분할된 부분메시지들의 크기가 동일하도록 패딩;
  return M'={MP1 ‖ MP2 ‖ … ‖ MPL};
  /* M'은 부분메시지들의 연접
     MPl={b1 ‖ b2 ‖ … ‖ bJ}
     J는 부분메시지의 크기 */
}
```

<그림 1> 전체메시지 분할 알고리즘

메시지 통합 방법은 부분메시지의 인덱스와 부분메시지를 구성하는 비트들의 인덱스 순서에 따라 부분메시지들을 L행 J열의 행렬로 가정하고, 행우선에 따라 요소들을 하나로 통합한다. <그림 2>는 이에 대한 알고리즘을 보여준다.

```
Message_Integration(M', L){

  if (부분메시지들에 패딩된 비트가 있는가)
    부분메시지들에 패딩된 비트 제거;
  for(j=1; j<=J; j++)
    for(l=1; l<=L; l++)
      M={M ‖ bjl};
  return M={b1 ‖ b2 ‖ … ‖ bI};
}
```

<그림 2> 부분메시지 통합 알고리즘

3.2. 다중서명 방식

① 키 생성

- 크기가 k비트인 두 개의 큰 소수 p_l과 q_l을 선택한 뒤, $n_l=p_lq_l$인 n_l을 생성한다.
- $\gcd(d_l, \phi(n_l))=1$인 d_l와 $e_ld_l \equiv 1 \pmod{\phi(n_l)}$인 e_l을 구한다.
- 개인키 (d_l, n_l)는 저장하고, 공개키 (e_l, n_l)는 공개한다.

② 다중서명 생성 및 검증

<서명자1>

- 전체메시지 분할 알고리즘을 이용하여 전체메시지를 부분메시지들로 분할한다.
- $S_1=MP_1^{d_1} \bmod n_1$과 같이 서명을 생성하여, 다음 서명자에게 $\{S_1 \| MP_2 \| \cdots \| MP_L\}$을 전송한다.

<서명자2>

- <서명자1>로부터 수신된 $\{S_1 \| MP_2 \| \cdots \| MP_L\}$에서 S_1을 저장한 뒤, $MP_1=S_1^{e_1} \bmod n_1$과 같이 복호화한다.
- 부분메시지 통합 알고리즘을 이용하여 전체메시지 M을 복원한다.
- $S_2=MP_2^{d_2} \bmod n_2$와 같이 서명을 생성하여, 다음 서명자에게 $\{S_1 \| S_2 \| MP_3 \| \cdots \| MP_L\}$을 전송한다.

<서명자$l(3 \le l \le L)$>

- <서명자$l-1$>로부터 수신된 $\{S_1 \| \cdots \| S_{l-1} \| MP_l \| \cdots \| MP_L\}$에서 $\{S_1 \| \cdots \| S_{l-1}\}$을 저장한 뒤, 각각의 공개키를 사용하여 복호화한다.
- 부분메시지 통합 알고리즘을 이용하여 전체메시지 M을 복원한다.
- $S_l=MP_l^{d_l} \bmod n_l$과 같이 서명을 생성하여, 다음 서명자에게 $\{S_1 \| \cdots \| S_l \| MP_{l+1} \| \cdots \| MP_L\}$을 전송한다.
- 만약 $l=L$이라면 $\{S_1 \| S_2 \| \cdots \| S_L\}$을 <검증자>에게 전송한다.

<검증자>

- <서명자L>로부터 수신된 $\{S_1 \| S_2 \| \cdots \| S_L\}$을 저장한 뒤, $MP_l=S_l^{e_l} \bmod n_l$과 같이 하여 복호화한다.
- 부분메시지 통합 알고리즘을 이용하여 전체메시지 M을 복원한다.
- 전체메시지 M과 다중서명 $\{S_1 \| S_2 \| \cdots \| S_L\}$을 함께 저장하고, 모든 서명자에게 다중서명이 검증되었음을 알린다.

4. 토론

4.1. 보안 문제

2.3.절의 제안된 메시지 분할을 이용한 RSA 다중서명 방식은 키 생성이 RSA 방식과 같다. 전체메시지를 분할하고 각 서명자가 그 할당된 부분메시지에 제각기 서명을 하지만, 서명방식도 RSA 방식을

따른다. 따라서 보안 강도는 RSA와 같다.

4.2. 분할 및 통합 알고리즘의 시간 복잡도

전체메시지 분할 알고리즘에는 1부터 J까지 반복하는 반복문이 하나 있다. 여기서 J는 전체메시지 M을 구성하는 비트열의 크기이므로 반복문은 전체메시지의 크기 $|M|$만큼 반복 수행한다. 따라서 시간 복잡도는 $O(|M|)$이다.

<그림 2>의 부분메시지 통합 알고리즘은 1부터 J까지의 반복문 안에 1부터 L까지의 반복문이 내포된 이중 반복문이 하나 있다. J는 부분메시지 MP_l을 구성하는 비트열의 크기이고, L은 서명자의 수이다. 따라서 시간 복잡도는 $O(L|MP|)$이고, $L|MP|=|M|$이므로 $O(|M|)$이다.

해당 부분메시지에 서명하기 전에 첫 번째 서명자는 전체메시지를 분할하고, 그 외의 모든 서명자는 부분메시지들을 통합한다. 따라서 제안된 메시지 분할을 이용한 RSA 다중서명 방식에서 모든 서명자는 다중서명 생성 및 검증 외에 추가적인 계산 비용으로 시간 복잡도가 $O(|M|)$인 함수를 실행한다.

4.3. 블록 수와 서명자 수의 관계

전체메시지의 블록 수를 B라 하고, 서명자 수를 L이라고 하며, m을 양의 정수라 하자.

제안된 메시지 분할을 이용한 RSA 다중서명 방식에서 $B=mL-1$이면 블록의 수가 1개 증가하고, $B=mL+1$이면 블록의 수가 $L-1$개 증가한다. 반면 $B=mL$이면 블록의 수는 증가하지 않는다. 따라서 본 방식은 $B=mL$일 때 가장 효율성이 좋으며, $B=mL+1$일 때 가장 효율성이 좋지 않다.

5. 평가

디지털 서명은 숫자로 변환된 메시지와 숫자로 된 키의 계산 결과이기 때문에 숫자이다[6]. 디지털 서명에 기반을 둔 다중서명도 숫자이므로 다중서명의 생성 및 검증 시간과 복호화 시간은 그에 소요되는 계산량과 같다. 모든 서명자의 키 길이가 비슷하다고 했을 때 계산량은 서명 대상인 메시지의 크기로 비교할 수 있고, 메시지의 크기는 블록 수로 나타낼 수 있다. 따라서 다중서명의 길이와 통신량과 마찬가지로 다중서명의 생성 및 검증 시간과 복호화 시간을 블록 수로 계산한다.

제안된 메시지 분할을 이용한 RSA 다중서명 방식은 블록의 수가 증가하지 않는 경우, 블록의 수가 $L-1$개 증가하는 경우, 그리고 증가하는 평균 블록의 수가 $\frac{L-1}{2}$인 평균의 경우로 구분하여 평가한다.

5.1. 다중서명 길이

RSA 원형 다중서명 방식은 전체메시지 L개의 복사본에 각 서명자가 서명을 하고 그 서명들의 연접을 다중서명으로 간주하기 때문에 다중서명의 길이가 LB이다.

RSA 변형 다중서명 방식은 하나의 전체메시지에 각 서명자가 반복하여 서명을 하기 때문에 다중서명의 길이가 B이다.

제안된 메시지 분할을 이용한 RSA 다중서명 방식은 전체메시지를 서명자의 수만큼 분할하여 각 서명자가 그 할당된 부분메시지에 서명한다. 따라서 다중서명의 길이가 블록의 수가 증가하지 않는 경우는 B이고, 블록의 수가 $L-1$개 증가하는 경우에는 $B+(L-1)$이며, 평균의 경우는 $B+(\frac{L-1}{2})$이다.

5.2. 다중서명의 생성 시간

첫 번째 서명자부터 시작하여 마지막 서명자까지 서명이 완료되면 다중서명이 생성된다. 다중서명 생성 시간은 각 서명자 단계에서 서명하여야 할 블록 수의 합으로 계산한다.

RSA 원형 다중서명 방식과 RSA 변형 다중서명 방식의 다중서명 생성 시간은 두 방식 모두 L명의 서명자가 모두 블록 수가 B인 전체메시지에 서명을 하기 때문에 $\sum_{l=1}^{L}B=LB$이다.

제안된 메시지 분할을 이용한 RSA 다중서명 방식의 다중서명 생성 시간은 분할된 각 부분메시지에 서명을 생성하는 데 걸리는 시간으로, 블록의 수가 증가하지 않는 경우는 $\sum_{l=1}^{L}\frac{B}{L}=B$이고, 블록의 수가 $L-1$개 증가하는 경우는 $\sum_{l=1}^{L}\frac{B+L-1}{L}=B+L-1$이며, 평균의 경우는 $\sum_{l=1}^{L}\frac{B+\frac{L-1}{2}}{L}=B+\frac{L-1}{2}$이다.

5.3. 다중서명의 검증 시간

두 번째 서명자부터는 이전 서명자의 서명을 검증할 필요가 있을 수도 있다. 다중서명의 검증 시간은 모든 서명자가 이전 서명자들의 서명을 검증한다는 가정 하에 검증자와 두 번째 서명자부터 마지막 서명자까지에서 검증하여야 할 서명된 블록 수의 합으로 계산한다.

RSA 원형 다중서명 방식과 RSA 변형 다중서명 방식의 다중서명 검증 시간은 $\sum_{l=1}^{L}lB=\frac{L}{2}(L+1)B$이다.

제안된 메시지 분할을 이용한 RSA 다중서명 방식의 다중서명 검증 시간은 블록의 수가 증가하지 않는 경우 $\sum_{l=1}^{L}l\frac{B}{L}=\frac{1}{2}(L+1)B$이고, 블록의 수가 $L-1$개 증가하는 경우 $\sum_{l=1}^{L}l\frac{B+L-1}{L}=\frac{1}{2}(L+1)B+\frac{1}{2}(L^2-1)$이며, 평균의 경우 $\sum_{l=1}^{L}l\frac{B+\frac{L-1}{2}}{L}=\frac{1}{2}(L+1)B+\frac{1}{4}(L^2-1)$이다.

5.4. 복호화 시간

첫 서명자를 제외한 $(L-1)$명의 서명자와 검증자는 원래의 전체메시지를 보기 위해 복호화를 필요로 한다. 복호화 시간은 검증자와 두 번째 서명자부터 마지막 서명자까지에서 복호화가 필요한 블록 수의 합으로 계산한다.

RSA 원형 다중서명 방식은 L개의 전체메시지 복사본에 서명자들이 각각 서명을 하였기 때문에 하

다중서명 방식 \ 비교대상		다중서명 길이	다중서명 생성 시간	다중서명 검증 시간	복호화 시간	통신량
RSA 원형 다중서명 방식		LB	LB	$\frac{L}{2}(L+1)B$	LB	$(\frac{L+1}{2})LB$
RSA 변형 다중서명 방식		B	LB	$\frac{L}{2}(L+1)B$	$\frac{L}{2}(L+1)B$	LB
제안된 메시지 분할을 이용한 RSA 다중 서명 방식	블록의 수가 증가하지 않는 경우	B	B	$\frac{1}{2}(L+1)B$	$\frac{1}{2}(L+1)B$	LB
	블록의 수가 $L-1$개 증가하는 경우	$B+(L-1)$	$B+(L-1)$	$\frac{1}{2}(L+1)B+\frac{1}{2}(L^2-1)$	$\frac{1}{2}(L+1)B+\frac{1}{2}(L^2-1)$	$LB+L(L-1)$
	평균의 경우	$B+(\frac{L-1}{2})$	$B+(\frac{L-1}{2})$	$\frac{1}{2}(L+1)B+\frac{1}{4}(L^2-1)$	$\frac{1}{2}(L+1)B+\frac{1}{4}(L^2-1)$	$LB+\frac{L(L-1)}{2}$

<표 1> 기존의 다중서명 방식과의 비교

나의 전체메시지 복사본만을 복호화하면 된다. 따라서 복호화 시간은 $\sum_{l=1}^{L}B=LB$이다.

RSA 변형 다중서명 방식에서 서명자들은 하나의 동일한 전체메시지에 반복 서명을 했기 때문에 이전 서명자가 서명한 만큼 반복하여 복호화를 하여야 한다. 따라서 복호화 시간은 $\sum_{l=1}^{L}lB=\frac{L}{2}(L+1)B$이다.

제안된 메시지 분할을 이용한 RSA 다중서명 방식의 복호화 시간은 블록의 수가 증가하지 않는 경우 $\sum_{l=1}^{L}lB=\frac{1}{2}(L+1)B$이고, 블록의 수가 $L-1$개 증가하는 경우 $\sum_{l=1}^{L}l\frac{B+L-1}{L}=\frac{1}{2}(L+1)B+\frac{1}{2}(L^2-1)$이며, 평균의 경우 $\sum_{l=1}^{L}l\frac{B+\frac{L-1}{2}}{L}=\frac{1}{2}(L+1)B+\frac{1}{4}(L^2-1)$이다.

5.5. 통신량

통신량은 첫 번째 서명자에서부터 검증자까지 전송되는 통신량의 합으로 계산하였다.

RSA 원형 다중서명 방식의 통신량은 $\sum_{l=1}^{L}lB=(\frac{L+1}{2})LB$이다.

RSA 변형 다중서명 방식의 통신량은 $\sum_{l=1}^{L}B=LB$이다.

제안된 메시지 분할을 이용한 RSA 다중서명 방식의 통신량은 블록의 수가 증가하지 않는 경우 $\sum_{l=1}^{L}B=LB$이고, 블록의 수가 $L-1$개 증가하는 경우 $\sum_{l=1}^{L}B+L-1=LB+L(L-1)$이며, 평균의 경우 $\sum_{l=1}^{L}B+\frac{L-1}{2}=LB+\frac{L(L-1)}{2}$이다.

<표 1>은 지금까지의 평가 결과를 보여준다.

6. 결론

본 논문에서는 RSA 방식과 관련된 기존의 다중서명 방식의 문제점을 분석하고, 그 문제점을 해결할 수 있는 다중서명 방식을 제안하였다.

제안된 메시지 분할을 이용한 RSA 다중서명 방식은 시간 복잡도가 $O(|M|)$인 계산 비용을 추가적으로 필요로 한다. 그러나 B가 커질수록 기존의 다중서명 방식 보다 효율적이다. 특히 블록의 수가 증가하지 않는 경우 기존 방식에 비해 다중서명의 길이는 증가하지 않으면서 다중서명의 생성 및 검증 시간은 각각 $\frac{1}{L}$로 감소하고, 통신량과 복호화 시간은 각각 약 $\frac{1}{2}\sim\frac{1}{L}$로 감소한다.

따라서 기존의 다중서명 방식보다 효율적이라고 할 수 있다.

참고문헌

[1] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public key cryptosystems”, ACM 21, pp. 120~126, 1978.

[2] Shun-Fu Pon, Erl-Huei Lu, and Jau-Yien Lee, “Dynamic reblocking RSA-based multisignatures scheme for computer and communication networks”, IEEE COMMUNICATIONS LETTERS, Vol. 6, No. 1, 2002.

[3] Tzong-Chen Wu, Chih-Chan Huang, D.-J. Guan, “Delegated multisignature scheme with document decomposition”, The Journal of Systems and Software 55, pp. 321~328, 2001.

[4] 강창구, 김대영, “디지털 다중서명 방식 비교”, 정보보호학회지, 1992.12

[5] L. Harn, T. Kiesler, “New scheme for digital multisignature”, Electrinics Letters 25, pp. 1002~1003, 1989.

[6] A. Menezes, P. van Oorschot, and S. Vanstone, “Handbook of Applied Cryptography”, pp. 425~488, 1996.