情報保護學會論文誌
第17卷 第4號, 2007. 8

URSA 애드혹 서명 알고리즘의 오류 수정

이 정 현†
삼성종합기술원

Fixing Security Flaws of URSA Ad hoc Signature Scheme

Jeong Hyun Yi†
Samsung Advanced Institute of Technology

요 약

애드혹 네트워크는 완전 분산형 네트워크 구조로 인해 자원 효율성, 확장 가능성, 결함 허용성 측면에서 많은 장점이 있는 반면에, 다른 한편으로는 완전 분산형 네트워크 토폴로지가 보안 서비스 설계에 많은 도전을 안겨주기도 한다. 더구나, 네트워크 노드가 언제든지 추가 또는 탈퇴가 가능한 동적 토폴로지 변화는 보안 메카니즘을 설계 하는데 있어 어려움을 더하게 한다. 따라서, 애드혹 네트워크에서의 보안 서비스는 확장 가능하고 결함 허용이 가능하면서 네트워크 노드의 멤버쉽이 수시로 변경 가능하도록 하는 방식으로 제공되어야 한다. 본 논문에서는 임계치 암호 기술을 활용하여 기존 CA의 기능을 네트워크에 참여하고 있는 노드들에게 분산시켜 네트워크 자체적으로 인증 서비스가 가능 하도록 하는 분산 인증 기술 알고리즘을 살펴본다. 그러던 중 최근에 제안된 RSA 기반 애드혹 서명 알고리즘인 URSA 알고리즘에 중요한 보안 오류들이 있음을 [5]에서 지적한 바 있는데, 본 논문에서는 이들 오류들의 원인을 밝혀내고 이를 수정한 새로운 알고리즘을 제안한다.

ABSTRACT

Ad hoc networks enable efficient resource aggregation in decentralized manner, and are inherently scalable and fault-tolerant since they do not depend on any centralized authority. However, lack of a centralized authority prompts many security-related challenges. Moreover, the dynamic topology change in which network nodes frequently join and leave adds a further complication in designing effective and efficient security mechanism. Security services for ad hoc networks need to be provided in a scalable and fault-tolerant manner while allowing for membership change of network nodes. In this paper, we investigate distributed certification mechanisms using a threshold cryptography in a way that the functions of a CA (Certification Authority) are distributed into the network nodes themselves and certain number of nodes jointly issue public key certificates to future joining nodes. In the process, we summarize one interesting report [5] in which the recently proposed RSA-based ad hoc signature scheme, called URSA, contains unfortunate yet serious security flaws. We then propose new scheme by fixing their security flaws.

Keywords : *Ad hoc networks, threshold cryptography, distributed PKI, verifiable secret sharing*

I. 서 론

본 논문에서는 애드혹 환경에 가장 적합한 최초의

접수일: 2007년 4월 7일; 채택일: 2007년 6월 1일
† 주저자, jeong.yi@samsung.com

RSA기반 임계치 서명(Threshold Signature) 알고리즘으로 알려진 URSA (Ubiquitous and Robust Security Architecture) 알고리즘[2][3][4]을 분석하고, 원저자들의 주장과는 달리, 단지 1개의 비정상노드에 의한 DoS (Denial of Service) 공격에 대해서도 프로토콜이 정상적으로 동작하지 못함을 지적하고, 이의 오류를 수정한 알고리즘을 제안한다. 또한, 제안 알고리즘의 성능 분석을 통하여 DSA기반 임계치 서명 알고리즘인 TS-DSA[5]보다 계산량 및 통신량 측면 모두에서 효율적임을 보여 준다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 URSA 알고리즘의 소개와 보안오류들을 지적하고, 3장에서는 발견된 오류들을 수정 보완한 새로운 알고리즘을 제안한다. 4장에서는 제안 알고리즘의 특성과 성능을 비교 분석하고, 5장에서 결론을 맺는다.

Ⅱ. URSA 알고리즘의 보안 오류

최근에 Luo, et al.이 다수의 논문[2],[3],[4]을 통해 애드혹 환경에 적용 가능한 분산 PKI (Public Key Infrastructure) 구현을 위한 RSA 기반 임계치 서명 알고리즘인 URSA를 제안하였는데, Narashimha, et al은 이 알고리즘의 보안오류를 [5]에서 지적한 바 있다. 본 장에서는 URSA 알고리즘을 간단하게 소개하고, [5]에서 지적한 제안 알고리즘이 정확성 검증기능 (및 오류추적 기능)을 제공하지 못함을 요약 정리한다.

2.1. 초기화

신뢰할 수 있는 딜러(trusted dealer)가 RSA 키 쌍을 생성한 후, Shamir의 secret sharing[1]을 위한 다항식 $f(z)$를 아래와 같이 생성한다.

$$f(z) = \sum_{k=0}^{t-1} a_k z^k \pmod N \tag{1}$$

이때, $f(0) = d$가 되도록 한다. 여기서 d는 RSA 비밀키 이고, N은 RSA 모듈로(modulus)이다. 그 다음, t개 이상의 초기 노드들(P_i)에게 쉐어(share) d_i와 서명(signature) s_i를 발급한다. 서명은 P_i의 공개키 인증서 (PKC_i) 발급에 필요한 인증서 요청 메시지 m_i를 대상으로 한다.

$$s_i = (m_i)^d \pmod N \tag{2}$$
$$d_i = f(id_i) \pmod N \tag{3}$$

또한, VSS (Verifiable Secret Sharing)[6]를 위한 witness들을 다음과 같이 계산하여 공개한다. 즉,

$$W_k = g^{a_k} \pmod N.$$

네트워크가 초기화된 후에는 딜러는 더 이상 필요로 하지 않는다.

2.2. 부분서명 생성 및 조합

신규노드(P_{n+1})가 네트워크에 조인하기 위해서는 자신의 공개키 인증서를 기존 노드들(P_i)로부터 발급 받아야 한다. 이를 위해 기존 노들이 신규노드의 인증서 요청 메시지 (m_{n+1})에 대한 다음과 같이 부분서명 (partial signature)을 생성하여 신규노드에게 전송한다.

$$s_{n+1}^{(i)} = (m_{n+1})^{d_i \lambda_i(0)} \pmod N \tag{4}$$

서로 다른 t개의 $s_{n+1}^{(i)}$ 수신한 신규노드는 이들 값들을 곱한 다음, t-bounded offsetting 알고리즘[4]을 적용하여 실제 서명 s_{n+1}를 획득한다.

2.3. 부분쉐어 생성 및 조합

신규노드가 네트워크에 조인한 후에, 자신도 나중에 다른 신규노드들의 부분서명을 발급해 주려면 자신의 비밀쉐어가 필요한데, 이를 위해 기존노드들이 다음과 같이 신규노드에게 부분쉐어(partial share) 생성하여 전달한다.

$$d_{n+1}^{(i)} = d_i \lambda_i(id_{n+1}) \pmod N \tag{5}$$

서로 다른 t개의 $d_{n+1}^{(i)}$를 단순하게 더하기만 하면 신규노드는 자신의 비밀쉐어 d_{n+1}를 구할 수 있다. 즉,

$$d_{n+1} = \sum_{i=1}^{t} d_{n+1}^{(i)} \pmod N.$$

참고로, $d_{n+1}^{(i)}$계산시, Lagrange 계수인 $\lambda_i(id_{n+1})$는 t개의 ID 정보만 있으면 누구나 계산할 수 있다. 따라서, 신규 노드가 $d_{n+1}^{(i)}$ 수신 후, 기존노드들의 비밀키인 d_i을 추출할 수 있기 때문에, 이를 방지하기 위해 $d_{n+1}^{(i)}$랜덤화

하여 전송하여야 하는데, 이를 위한 기법을 PSRS (Parital Share Random Shuffling) 이라 부른다[7].

2.4. 정확성 검증 오류

신규노드가 s_{n+1}와 d_{n+1}을 획득한 후, 이의 정확성을 검증하여야 하는 데, s_{n+1}의 검증은 시스템 공개키 (e, N)으로 표준 RSA 서명 검증 과정을 거치면 된다. d_{n+1}의 검증의 경우, VSS 기법을 적용하여야 하는데, 여기서 URSA 알고리즘의 문제점이 노출된다.

$$g^{d_{n+1}} \neq \prod_{k=0}^{t-1} (W_k)^{(id_{n+1})^k} \pmod N \tag{6}$$

즉, URSA 알고리즘은 저자들의 주장과 달리, 위의 검증성 테스트를 만족하지 못한다. 다시 말하면, 정상적으로 계산된 d_{n+1}와 (공격자에 의한 조작 또는 통신상의 오류 등으로 인한) 비정상적인 d'_{n+1}를 구분하지 못한다. 좀더 정확히는 두 값도 모두 항상 비정상적인 값으로만 신규노드가 받아들이게 된다.

Ⅲ. URSA 알고리즘의 오류수정

본 장에서는 URSA 알고리즘의 정확성 검증 오류의 원인을 살펴보고, 이의 수정방법과 오류추적 메카니즘을 제안한다.

정확성 검증 오류의 원인: 수식 (1)과 (5)에서 d_i와 $d_{n+1}^{(i)}$ 값들이 모듈로 $\phi(N)$이 아닌 N에 의해 계산되어졌기 때문에 일반적인 VSS 기법[6]을 이용하여 d_i의 정확성을 검증할 수가 없게 된다. 또한, $\phi(N)$는 초기화 때 신뢰할 수 있는 딜러에게만 알려지는 비밀값이므로 네트워크 초기화 후에는 어떤 노드에게도 알려지지 않는다. 따라서, 부분 쉐어 계산시에 기존노드가 $d_{n+1}^{(i)} = d_i \lambda_i (id_{n+1}) \pmod{\phi(N)}$와 같이 계산할 수도 없다.

오류 수정: 위의 오류를 수정하기 위해서는 쉐어들을 유한체 Z_N상에서 계산하는 것이 아니라, 일반 정수 상에서 계산함으로써 해결할 수 있다. 즉, 상기 수식 (1)과 (5)를 아래의 수식 (7)과 (8)로 각각 변경하면 쉐어의 정확성 검증과 오류추적은 정상적으로 동작하게 된다. 부분 서명의 생성과 오류추적 방법에 대해서는 다음 절에서 별도로 설명한다.

$$f(z) = \sum_{i=0}^{t-1} a_i z^i \tag{7}$$

$$d_{n+1}^{(i)} = d_i \lambda_i (id_{n+1}) \tag{8}$$

3.1. 부분서명 생성 및 조합

부분서명 생성은 2.3절의 방법과 유사하지만 Lagrange 계수 $\lambda_i(0)$을 계산에 포함하지 않는다. 대신 $\lambda_i(0)$는 부분서명 조합시에 신규노드가 계산하도록 한다. 즉,

$$s_{n+1}^{(i)} = (m_{n+1})^{d_i} \pmod N \tag{9}$$

그런 다음 부분서명의 단순 곱셈만으로도 원래 서명값을 구할 수다. 참고로, 아래 수식의 $\lambda_i(0)$도 정수 상에서 계산한다.

$$\prod_{i=1}^{t} (s_{n+1}^{(i)})^{\lambda_i(0)} = (m_{n+1})^d \pmod N \tag{10}$$

기존 URSA 알고리즘과의 차이점을 살펴보면, 부분서명 생성에 사용된 기존노드들의 쉐어를 정수 상에서 계산하였기 때문에, 최대 t번의 추가 지수승 연산이 요구되는 t-bounded offsetting 알고리즘의 적용없이 부분서명들의 단순 곱셈만으로 실제 서명을 구할수 있다. 또한, $\lambda_i(0)$ 계산을 기존 노드들이 부분서명 생성시에 포함한 것과는 달리, 신규노드가 서명 조합시에 포함하도록 변경함으로써, 토폴로지 수시로 변화는 애드혹 네트워크에서 정확히 t개의 기존 노드들이 동시에 $\lambda_i(0)$계산에 참여야 하는 단점인 Interaction을 제거할 수 있게 된다.

3.2. 오류 추적

부분쉐어 추적은 VSS 기법을 적용하면 쉽게 확인 할 수 있지만, 부분서명의 경우에는 VSS 기법의 단순 적용으로 해결되지 못한다. 따라서 이를 위해 Schnorr 서명[9]에 기반한 다음의 ZKP (Zero Knowledge Proof) 프로토콜 제안한다.

3.2.1. 부분서명 생성

2.2절에서 생성된 부분서명 과정과 함께 다음의 절차

가 추가적으로 필요하다.

(1) $s_{n+1}^{(i)}=(m_{n+1})^{d_i} \pmod N$을 계산한다.
(2) 랜덤 비밀값 $r \in Z_N^*$을 생성한다.
(3) $u=g^r \pmod N$을 계산한다.
(4) $v=(m_{n+1})^r \pmod N$을 계산한다.
(5) $c=H(s_{n+1}^{(i)}, u, v)$을 계산한다.
(6) $z=d_i c+r$을 계산한다.
(7) $(s_{n+1}^{(i)}, c, z)$를 신규노드에게 전송한다.

3.2.2. 부분서명 검증

신규 노드의 부분서명 검증과정은 다음과 같다.

(1) Witness값들을 이용하여 다음과 같이 w'를 계산한다.

$$w' = \prod_{k=0}^{t-1} (W_k)^{(id_i)^k} \pmod N$$

여기서, $w' = \prod_{k=0}^{t-1} (W_k)^{(id_i)^k} = g^{d_i} \pmod N$이 된다.

(2) $u'=g^z(w')^{-c} \pmod N$을 계산한다.
여기서, $u'=g^z(w')^{-c}=g^r \pmod N$이 된다.
(3) $v'=(m_{n+1})^z(s_{n+1}^{(i)})^{-c} \pmod N$을 계산한다.
여기서, $v'=(m_{n+1})^z(s_i)^{-c}=m_{n+1})^r$ 이 된다.
(4) $c'=H(s_{n+1}^{(i)}, u', v')$을 계산한다.
(5) $c=c'$ 를 비교한다.

위 검증과정에서 단계 (2)와 (3)을 모두 거쳐야 하는 이유는 다음과 같다.

1) u'만으로 검증할 경우, d_i의 정확성은 검증할 수 있으나, 부분서명 생성에 올바른 d_i가 사용되었는지를 확인할 수 없게 된다.
2) 부분서명 $s_{n+1}^{(i)}$의 위변조 유무를 v'만으로 검증할 경우, 공격자가 임의의 키 $\tilde{d}_i$로 위조된 서명 $\tilde{s}_{n+1}^{(i)}=(m_{n+1})^{\tilde{d}_i}$을 생성하고, 또 다른 랜덤값 $\tilde{r}$으로 $\tilde{v}=(m_{n+1})^{\tilde{r}}$을 계산하여 검증자에게 전달할 경우, 검증자는 위조서명을 정상으로 받아들이게 된다.

〔표 1〕 제안 알고리즘의 특성 비교

구분	TS-DSA	URSA	제안 알고리즘
서명생성방식	Interactive	Interactive	Non-interactive
쉐어생성방식	Interactive	Interactive	Interactive
정확성 검증기능	제공	제공 못함	제공
오류추적 기능	제공	제공 못함	제공
부분쉐어 램덤화	요구됨	요구됨	요구됨

Ⅳ. 비교 분석

본 장에서는 수정한 알고리즘의 특성과 성능을 DSA 기반 임계치 서명 알고리즘인 TS-DSA[5]와 비교한다.

〔표 1〕에서는 제안 알고리즘의 특성을 TS-DSA 뿐만 아니라 기존 URSA와도 비교한다. TS-DSA는 서명 및 쉐어 생성시에는 DSA 알고리즘 자체의 randomness 성질과 Lagrange 계수 계산을 위해 t개의 기존 노드가 동시에 참여하여야 하는 Interaction이 필요로 한 반면, 제안 알고리즘을 이용한 서명생성 시에는 이러한 Interation을 필요로 하지 않는다.

〔표 2〕는 제안 알고리즘의 성능을 TS-DSA와 계산 및 통신 복잡도 측면에서 비교 분석한다. 좀더 구체적으로 설명하면, 서명연산과 관련하여 제안알고리즘의 경우 t개의 부분서명과 3.2.1절에서 알수 있듯이 부분서명의 오류추적을 위한 추가 연산으로 2t번의 지수승을 필요로 하므로, 서명 생성을 위해 총 3t번의 지수승 연산이 필요하고, 검증은 표준 RSA 검증을 하므로 지수승

〔표 2〕 TS-DSA과 제안 알고리즘의 성능 비교

구분			TS-DSA	제안 알고리즘
계산 복잡도 (지수승수)	서명 연산	생성	$4t-1$	$3t$
		검증	2	1
		추적	$2t^2+5t-3$	t^2+4t
	쉐어 연산	생성	t^2-t	t^2-t
		검증	t	t
		추적	t^3-t^2+3t	t^3-t^2+3t
통신 복잡도	라운드 수	broadcast	$3t$	$t+1$
		unicast	$5t^2+3t-3$	t^2+2t
	대역폭 (비트수)		$(9t^2-7t+1)\log p+$ $(9t^2-8t+2)\log q$	$2t^2\log N$

이 1회 필요로 한다. 또한 3.2.2절의 부분 서명 추적이 필요할 경우, t^2+4t번의 지수승이 필요로 하게 된다. 쉐어 연산과 통신 복잡도의 경우, 대부분의 연산이 PSRS[7]때문에 발생하는 것으로 자세한 내용은 [7]을 참조하길 바란다. TS-DSA의 경우는 알고리즘 특성상 최소 2t-1개의 부분서명과 부분쉐어가 필요로 하는데, 자세한 내용은 [5]를 참조하면 된다.

[표 2]의 분석 결과에서 알 수 있듯이, 제안 알고리즘이 서명 생성, 정확성 검증, 오류추적 모든 면에서 TS-DSA 보다 효율적이며, 쉐어 연산은 동일한 복잡도를 나타낸다. 또한, 제안 알고리즘이 TS-DSA보다 통신 오버헤드를 대폭 줄일 수 있음을 알 수 있다. 특히, 대략 4-5배 정도($\log p \cong \log N$로 가정)의 대역폭 감소는 무선통신에서 데이터 전송량이 네트워크 노드의 전력소모량과 아주 밀접한 관련이 있으므로[8], 제안 알고리즘이 배터리 기반의 무선망 장비들의 전력 소비량 감소에 많은 기여를 할 수 있을 것으로 기대된다.

V. 결 론

본 논문에서는 최근에 제안된 RSA 기반 애드혹 서명 알고리즘인 URSA의 분석을 통하여, 이 알고리즘의 보안 오류가 있음을 지적하였다. 또한 오류의 원인을 파악한 후, 이를 수정한 새로운 알고리즘을 제안하였다. 제안한 알고리즘은 기존 DSA 기반 애드혹 서명 알고리즘인 TS-DSA 보다 계산량 및 통신량 측면에서 보다 효율적임을 보였다. 제안 알고리즘을 통해 발급되는 인증서의 서명은 표준 RSA 서명과 호환이 되기 때문에 DSA 보다 훨씬 빠른 검증속도를 가진 RSA의 장점을 그대로 살릴 수 있다.

참고문헌

[1] A. Shamir, "How to Share a Secret", *Communications of the ACM*, 22(11), 1979.

[2] J. Kong, H. Luo, K. Xu, D. L. Gu, M. Gerla, and S. Lu, "Adaptive Security for Multi-level Ad-hoc Networks", *Journal of Wireless Communications and Mobile Computing*, volume 2, pp.533-547, 2002.

[3] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing Robust and Ubiquitous Security Support for MANET", *IEEE International Conference on Network Protocols (ICNP'01)*, pp.251-260, 2001.

[4] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, "URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks", *IEEE/ACM Transactions on Networking*, 12(6), pp.1049-1063, 2004.

[5] M. Narasimha, G. Tsudik, and J. H. Yi, "On the Utility of Distributed Cryptography in P2P and MANETs: the Case of Membership Control", *IEEE International Conference on Network Protocols (ICNP'03)*, pp.336-345, 2003.

[6] P. Feldman, "A Practical Scheme for Non-interactive Verifiable Secret Sharing", *Symposium on Foundations of Computer Science (FOCS'87)*, pp.427-437, 1987.

[7] T. P. Pedersen, "A Threshold Cryptosystem without a Trusted Party", *Eurocrypt'91*, LNCS No. 547, pp.522-526, 1991.

[8] K. Barr and K. Asanovic, "Energy Aware Lossless Data Compression", *International Conference on Mobile Systems, Applications, and Services (MobiSys'03)*, May 2003.

[9] C. P. Schnorr. "Efficient Signature Generation by Smart Cards", *Journal of Cryptology*, Vol. 4, No. 3, pp.161-174, 1991.

〈著者紹介〉

이 정 현 (Jeong Hyun Yi) 정회원

1993년 2월 : 숭실대학교 전자계산학과 학사
1995년 2월 : 숭실대학교 전자계산학과 석사
2005년 8월 : Univeristy of California at Irvine, Information and Computer Science 박사
1995년 2월~2001년 7월 : 한국전자통신연구원 연구원
2000년 4월~2001년 3월 : 미국 표준기술연구원(NIST) 객원연구원
2005년 10월~현재 : 삼성종합기술원 수석연구원
<관심분야> 네트워크 보안, 분산시스템 보안, 암호응용