암호 어큐뮬레이터를 이용한 온라인 공개키 인증서 유효성 확인 방법

김재형*, 조정식*, 김순석**, 김성권*

*중앙대학교 컴퓨터공학과, **한라대학교 정보통신공학부

On-line Authentication of Public-Key Certification Using Cryptographic Accumulator

Jae-hyung Kim*, Jung-sik Cho*, Soon-seok Kim**, Sung-kwon Kim*

*Department of Computer Science & Engineering Chung-Ang Univ.
**School of Information & Communication Engineering Halla Univ.

요 약

본 논문은 공개된 네트워크를 통해 교환되는 기밀정보의 무결성과 인증을 위해 사용되는 공기키 인증서에 대한 새로운 인증방법을 제안한다. 제안된 새로운 방법은 인증된 딕셔너리[1] 기반하에 RSA 일방향 어큐뮬레이터를 이용하여 인증서 폐기목록(CRL: Certificate Revocation List) 원소들의 존재 유무에 대한 증거값들을 미리 생성하고 사용자가 인증서의 유효성을 알아보기 위해 해당 원소에 대한 질의를 보내게 되면 미리 계산되어진 증거값을 바로 제시해줌으로써 계산시간이나 전송용량의 효율성을 극대화 시켰다. 또한 파라미터값을 주어 CRL을 여러개의 부분집합으로 나눔으로써 계산량을 동적으로 분산시켜 사용자가 소형기기나 무선 네트워크 환경에 있더라도 활용이 가능하도록 하였다.

I. 서론

최근 무선인터넷 보급 확대와 전자상거래 증가, 이동성 확대, 인터넷 사용인구 증가 등으로 핸드헬드PC시장이 급속도로 증가하고 있으며 PDA를 이용한 증권거래 시스템과 이동통신회사들의 휴대전화를 통한 전자상거래 결제시스템이 서비스를 시작함으로써 소형기기를 이용한 전자상거래 및 전자금융거래가 활성화되고 있다. 따라서 유·무선 인터넷과 같은 공개된 네트워크에서 엄격한 보안이 요구되는 기밀정보가 교환되고 있으며 이러한 기밀정보의 안전한 교환을 위하여 공개키 인증서가 사용되고 있다. 그러나 현재 사용되고 있는 공개키 기반구조(PKI: Public-Key Infrastructure)에서 공개키 인증서를 인증하기 위한 CRL의 관리는 사용자의 계산량과 데이터 전송량이 매우 크다는 문제점을 갖고 있다. 또한 사용자의 증가로 인한 네트워크 과부하도 간과할 수 있는 문제는 아닐 것이다. 이러한 문제점들을 개선하기 위해 인증된 딕셔너리라는 자료구조를 이용하여 네트워크 과부하를 막을 수 있는 공모 가능한 미러사이트들을 안전하게 운영하여야 하고 일방향 어큐뮬레이터를 이용하여 공개키 인증서의 인증을 위한 증거값을 미리 만들어 놓음으로써 사용자가 최소의 계산량으로 증거값을 검증할 수 있도록 하여야 하며 또한 전송되는 데이터도 그 크기를 최소화해야 할 것이다.

II. 본문

1. 관련 연구

본 절에서는 제안하는 방법에 사용되는 중요한 암호학적 개념에 대해 설명하고자 한다.

1) 일방향 어큐뮬레이터

일방향 어큐뮬레이터[2,3,4,5]는 제안하는 방법에서 가장 중요한 도구이다. 이 방법의 가장 일반적인 형식은 초기값 y_0를 시작으로 원소들의 집합 $X=\{x_1, x_2, \ldots, x_n\}$에 대해 일방향 함수 f를 이

용하여 $y_i = f(y_{i-1}, x_i)$값을 구해나가는 것이다. 일방향 어큐뮬레이터 함수의 잘 알려진 예제로는 지수승 어큐뮬레이터가 있다. 지수승 어큐뮬레이터는 적절히 선택된 y_0와 계수 N이 필요하다. 선택된 N값은 두 개의 큰 소수 p와 q의 곱을 나타내는 것으로 RSA 암호법에 사용되는 계수이다. 이러한 어큐뮬레이터를 수식으로 표현하면 $\exp(y, x) = y^x \bmod N$ 이며 다음절에서 이 어큐뮬레이터가 인증된 딕셔너리에서 어떻게 사용되는지 보여준다.

2) 딕셔너리

일반적으로 딕셔너리는 믿을 수 있는 정보제공자, 믿을 수 없는 디렉토리 그리고 사용자로 표현해 볼 수 있다. 믿을 수 있는 정보제공자는 추가하거나 삭제할 수 있는 원소들의 집합 S를 정의하고 믿을 수 없는 디렉토리는 정보제공자로부터 받은 집합 S의 복사본을 유지하며 사용자의 질의에 대한 특정원소의 존재여부를 정보제공자가 서명한 인증정보를 포함하여 예/아니오로 응답한다. 정보제공자와 디렉토리가 집합 S에 대해 질의와 업데이트 프로토콜을 유지하는데 사용하는 자료구조를 인증된 딕셔너리[1]라고 부른다.

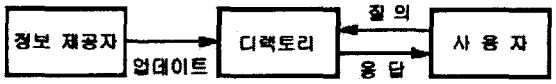

그림 1: 인증된 딕셔너리

2. 기존 연구

1) Goodrich 방법

가. 스트레이트 포워드 방법

스트레이트 포워드 방법[6]은 집합 $S=\{e_1, e_2, \ldots, e_n\}$를 정보제공자가 가지고 있는 정보들의 집합이라 하고, 정보제공자는 서로 다른 큰소수 p, q를 선택하여, $N = pq$를 계산하고 N과 서로소인 밑수 a를 고른다. 또한 원소 e_i에 대해 $h(x_i) = e_i$인 식별자 x_i를 이용해 $A = a^{x_1 x_2 \cdots x_n} \bmod N$ 과 타임스탬프 t를 묶어 메시지 (A, t)를 모든 디렉토리에 전송한다. 이러한 초기화 과정이 끝나고 사용자가 원소 x_i가 집합 S에 속했는지에 대한 질의를 디렉토리에게 보내면 $A_i = a^{x_1 x_2 \cdots x_{i-1} x_{i+1} \cdots x_n} \bmod N$ 값을 계산하여 $A_i, N, (A, t)$를 사용자에게 되돌려 준다. 사용자는 디렉토리로부터 받은 메시지 중 A_i에 x_i승을 해줌으로서 $A_i^{x_i} = A$임을 확인한다. 정보제공자는 집합 S에 대해 새로 추가되거나 삭제되는 정보가 있을 경우 디렉토리에게 업데이트정보를 보내어 추가되는 경우는 기존의 A에 추가되는 정보 x_j승을 하여 새로운 A를 생성하고 삭제되는 경우는 해당되는 정보를 빼고 다시 A를 생성한다.

나. 전처리 어큐뮬레이터

위의 스트레이트포워드 방법은 디렉토리가 질의를 받아서 해당되는 A_i를 만들어야하기 때문에 질의에 대한 응답시간이 늦어진다. 전처리 어큐뮬레이터[6]는 이러한 단점을 보완하기 위해 다음의 두단계를 수행된다.

첫 번째 단계는 전 이진트리 T를 구성하여 노드 v가 종단 노드일 경우 $x(v) = x_i \bmod \Phi(N)$로 계산하고 v가 내부노드일 경우는 왼쪽 자식노드 l와 오른쪽 자식노드 r에 대해 $x(v) = x(l)x(r) \bmod \Phi(N)$를 계산한다. 두 번째 단계는 계산된 전 이진트리 T에서 노드 v에 있어서 부모노드 z와 형제노드 w를 이용하여 $A(v) = A(z)^{x(w)} \bmod N$을 계산한다. 이렇게 하면 사용자의 질의에 미리 계산된 A_i값을 보낼 수 있게 되어 응답시간을 빠르게 할 수 있다.

다. 파라미터를 이용한 어큐뮬레이터

이 방법은 파리미터값 p를 이용하여 집합 S를 p개의 그룹으로 나누어 계산하는 방법[6]으로 p값은 $1 \le p \le n$의 범위 내에서 정보제공자와 디렉토리의 계산 능력에 따라 유동적으로 조절될 수 있다는 것이 장점이다.

2) Faldella 방법

앞서 살펴본 Goodrich 방법은 원소 e_i가 집합 S에 속해 있지 않을 경우에 대한 증거값을 위해 추가적인 트리[7]를 구성해야하는 반면에 Faldella 방법[8]은 원소 e_i에 대해 각각의 식별자값 대신에 원소들 사이의 범위값인 Statement를 유지하여 $[e_i, e_{i+1}]$을 SHA-1 해쉬함수로 해쉬한 160bit 해쉬값에 1bit를 덧붙여 161bit 식별자를 사용함으로써 원소가 집합에 속하거나 그렇지 않을 경우 모두 한번의 응답으로 증명이 가능하게 한다.

3. 제안하는 방법

우선 전체 n개의 원소를 갖는 집합 $S=\{e_i \mid i=1\sim n, \text{단 } e_1 < e_2 < \ldots < e_n\}$를 p개의 부분집합 $B_1, B_2, \ldots, B_p$로 나눈다. 다음으로 n/p개의 원소를 갖는 각각의 부분집합 B_j에 대해 그 원소들의 범위값인 $[e_i, e_{i+1}]$의 집합인 Statement를 생성한다. 생성된 Statement의 원소

들인 범위값들은 각각 160-bit SHA-1 해쉬함수를 통해 해쉬된 후 마지막에 1을 덧붙여 홀수인 161bit 정수값의 식별자 x_i를 생성하게 된다.

가. 초기화

각각의 부분집합 B_j들의 원소들에 대한 식별자 x_{ji}가 생성되면 종단노드들에 부분집합의 원소들을 배열하고 종단노드일 경우 $x(v) = x_i \bmod \Phi(N)$, 내부노드일 경우는 자식노드 l와 r를 이용하여 $x(v) = x(l)x(r) \bmod \Phi(N)$를 계산하여 트리 T_j를 구성하고 부모노드 z와 형제노드 w를 이용하여 $C(v) = C(z)^{x(w)} \bmod N$을 계산한다. 이렇게 계산되어진 C_{ji}들과 각 부분집합의 증거값과 타임스탬프 t를 묶은 $(C_1, t), (C_2, t), \ldots, (C_p, t)$ 값들을 디렉토리에 전송하고 디렉토리는 정보제공자가 모두 계산하여 보내준 값을 저장한다.

나. 질의 및 확인

사용자는 본인이 갖고 있는 공개키 인증서가 폐기목록인 집합 S에 속하는지를 확인하기 위하여 디렉토리에게 원소 e_i대한 질의를 보내고 디렉토리는 e_i가 포함되는 부분집합 B_j를 찾고 식별자 x_{ji}를 생성하여 증거값 (C_j, t)값에 미리 계산되어진 C_{ji}값과 Statement 값을 함께 해당 사용자에게 보내준다. 사용자는 C_{ji}값에 x_{ji}승하여 C_j와 일치하는지 확인하고 Statement 에 e_i가 포함되는지 확인한다. 이때 p개의 부분집합에 속한 각각의 원소들의 증거값이 미리 계산되어 있기 때문에 기존의 방법에 비해 매우 빠른 응답을 할 수 있다. 또한 전송된 Statement에서 원소 e_i가 범위값에 포함되는지 아니면 범위값의 맨 앞에 있는지만을 판단하여 원소 e_i가 집합 S에 속해 있는지, 아닌지를 확인시켜줄 수 있다.

다. 업데이트

원소가 삽입될 경우는 정보제공자가 자신이 갖고 있는 정렬된 p개의 부분집합 중에서 해당원소가 포함되는 부분집합을 찾아서 Statement를 갱신하고 새로운 식별자를 생성하여 해당되는 트리만 재구성하면 된다. 원소가 삭제될 경우에는 삭제될 원소를 포함하고 있는 부분집합에서 해당원소를 삭제한 후 Statement를 갱신하여 식별자를 구하고 그 부분집합의 트리만을 재구성하면 된다. 이는 기존방법이 전체 원소를 포함하는 트리를 재구성하는데 비해 매우 적은 계산량이 요구된다.

라. 이론적인 성능 비교

기존에 일방향 어큐뮬레이터를 이용한 방법들은 모두 사용자에게 적은 계산량을 할당하도록 설계되었으나 질의에 대한 응답이나 업데이트에 필요한 계산량이 상대적으로 많아지는 단점이 있다. 또한 미리계산된 지수승 테이블이 너무 많은 저장공간을 차지하는 문제점도 있다. 제안하는 방법은 이러한 단점들을 보완한 방법으로 그 이론적인 성능을 비교하면 다음과 같다.

우선 질의에 대한 응답시간을 비교해보면 제안하는 방법은 $O(1)$로 Goodrich방법의 $O(n/p)$와 비교하였을 때 빠르게 실행된다. 또한 삽입, 삭제와 같은 업데이트 시간은 파라미터를 이용한 어큐뮬레이터가 $O(p + \log(n/p))$이고 제안하는 방법이 $O(n/p)$으로 p값의 변화에 따라 유동적이지만 역시 제안하는 방법이 빠름을 알 수 있다. 단 Faldella방법은 질의에 대한 응답시간이나 업데이트 시간이 모두 $O(1)$로 매우 빠르게 나타난 반면 차지하는 공간면에서 지수승 테이블을 유지해야 하기 때문에 $O(n + 2^m)$ (m은 N의 bit 수)공간을 차지하게 된다. 또한 시간 계산값에 적용되지 않는 부수적인 계산들이 많기 때문에 실질적인 수행시간은 이론과 다르다. 이는 4절 시뮬레이션 및 성능분석에서 자세히 보여준다.

4. 시뮬레이션 및 성능분석

아래 그림 2,3,4에서 x축은 p값의 변화를 나타낸 것이고 y축은 주어진 연산을 수행하는데 걸리는 평균시간을 microseconds 단위로 나타낸 것이다.

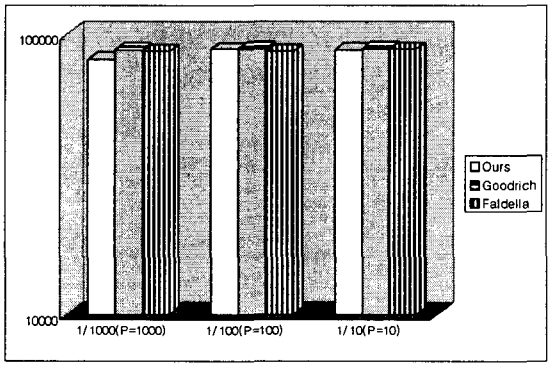

그림 2: 증거값 검증시간 비교

그림 2는 사용자가 디렉토리에게 보낸 질의에 대한 응답을 받아 해당 원소가 Statement에 속하는지 확인하고 $A_i^{x'} \bmod N = A$를 계산하여 해당 원소가 폐기되었지 또는 그렇지 않은지를 확인하고 질의에 대한 증거값을 검증하는데 걸린 시간을 측정한 것이다. 기존의 방법과 새로 제안하는 방법 모두 $O(1)$값을 갖고 있으며 실제 실험 결과도 0.1초 정도로 비슷한 결과를 보이고 있다.

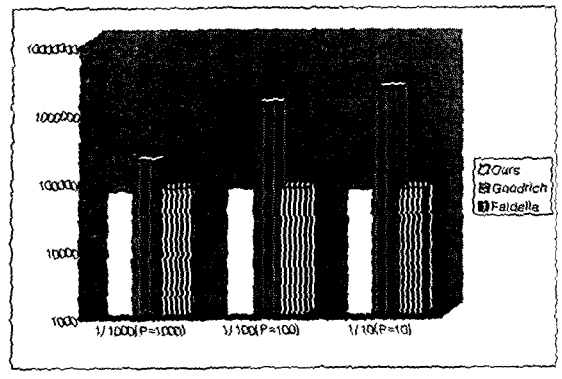

그림 3: 질의에 대한 응답시간 비교

그림 3에서 보듯이 제안하는 방법은 p값의 변화와 상관없이 $O(1)$을 유지하고 있는 반면에 *Goodrich*방법은 p값이 감소함에 따라 $O(1)$에서 $O(n)$로 증가로 증가하고 있으며 이는 실제 실험상에서도 유사하게 나타나고 있다. Faldella의 방법은 전체 원소를 p개의 부분집합으로 나누지 않고 어큐뮬레이터 계산을 빠르게 실행시키기 위한 미리 계산된 테이블을 사용하기 때문에 p값의 변화에 상관없이 일정하게 유지된다. 그러나 원소들의 곱을 구하는 시간이 실제 실행시간의 대부분을 차지하고 있어 그 실행시간은 비교적 느리다.

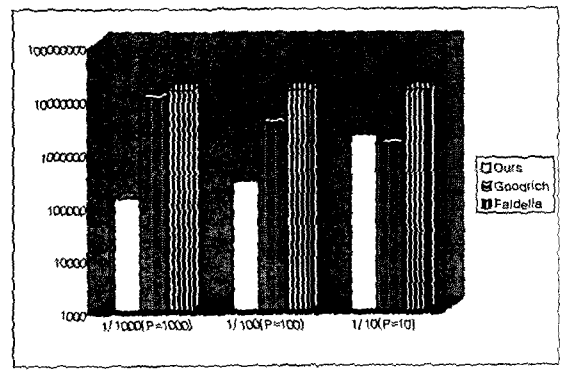

그림 4: 업데이트 시간 비교

그림 4는 새로운 원소를 업데이트 하는데 걸린 시간을 측정한 값이다. Faldella의 방법은 역시 전체 원소를 p개의 부분집합으로 나누지 않기 때문에 일정한 값을 유지하지만 원소들의 곱셈 연산이 실행시간을 지연시키고 있다. Goodrich방법은 p값이 작아짐에 따라 $O(n)$에서 $O(\log n)$로 연산시간이 줄어들고 있으며 제안하는 방법은 p값이 작아짐에 따라 $O(1)$에서 $O(n)$로 증가하는 것을 볼 수 있다. 그러나 Goodrich방법은 원소의 식별자를 계산시간이 45*millisecons*로 제안하는 방법의 4milliseconds보다 약 10배 가량 크기 때문에 실제 연산시간은 제안하는 방법에 비해 비효율적이다.

III. 결론 및 향후연구

위 시뮬레이션 결과에서 보듯이 제안하는 방법이 열악한 네트워크 환경을 갖는 소형기기에서는 더욱 안정적으로 동작할 수 있으며, 이러한 소형기기들이 실제 전자상거래나 전자금융거래를 이용하는데 매우 유용하게 쓰일 수 있음을 보였다.

참고문헌

[1] M. Naor and K. Nissim. Certificate *revocation and certificate* update. *In* Proceedings of the 7th USENIX Security Symposium (SECURITY-98), pages 217-228, *Berkeley, 1998.*

[2] J. Benaloh and M. de Mare. One-way accumulators: A decentralized alternative to *digital* signatures. *In* Advances in Cryptology-EUROCRYPT 93, *volume* 765 of Lecture *Notes in Computer* Science, pages 274-285, 1995.

[3] N. Baric and B. Pfitzmann. Collision-free accumulators *and fail-stop* signature schemes without trees. In Advances in Cryptology: Proc. EUROCRYPT, *volume* 1233 of Lecture Notes *in* Computer Science, pages 480-494, 1977.

[4] R. Gennaro, S. Halevi, and T. Rabin. *Secure* hash-and-sign signatures without the random oracle. In Advances in Cryptology: Proc. EUROCRYPT, volume 1592 of Lecture Notes in Computer Science, pages 123-139. Springer-Verlag, 1999.

[5] P. C. Kocher. On certificate revocation and validation. In Proc. International Conference on Financial Cryptography, volume 1465 of Lecture Notes in Computer Science, 1998.

[6] M. T. Goodrich, R. Tamassia and J. Hasic. An Efficient *Dynamic* and Distributed *Cryptographic* Accumulator. Johns Hopkins Information Security Institute, 2002.

[7] P. C. Kocher. *On certificate revocation* and validation. *In* Proc. International Conference on *Financial* Cryptography, volume 1465 of Lecture Notes in Computer Science, 1998.

[8] E. Faldella. A Flexible Scheme for On-Line Public-key Certificate Status Updating and Verification, Proceedings of the Seventh International Symposium on Computers *and Communications (ISCC'02)*