인증된 딕셔너리에서의 향상된 전처리 암호 어큐뮬레이터

김재형*○ 김순석** 김성권*
중앙대학교 컴퓨터공학과* , 한라대학교 정보통신공학부**
gckim○@alg.cse.cau.ac.kr , sskim@halla.ac.kr , skkim@cau.ac.kr

An Advanced Precomputed Cryptographic Accumulator in the Authenticated Dictionary

Jae-Hyung Kim*○ Soon-Seok Kim** Sung-Kwon Kim*
Dept. of Computer Science & Engineering, Chung-Ang University* ,
School of Information & Communication Engineering, Halla University**

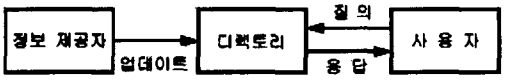
요 약

본 논문은 기존의 RSA 일방향 가산기를 이용한 인증된 딕셔너리[1]에 대한 개선된 방법을 제안한다. 제안된 새로운 방법은 신뢰할 수 있는 정보 제공자와 신뢰할 수 없는 디렉토리 그리고 사용자로 이루어진 그룹에서 정보 제공자가 디렉토리를 통해 제공하는 정보에 대해서 디렉토리는 검증된 정보만을 사용자에게 제공할 수 있도록 해주며 이러한 일련의 과정에서 일어나는 정보의 업데이트나 질의 그리고 검증이 효율적으로 이루어 질수 있도록 개선하였다. 이러한 연구는 PKI(Public Key Infratructure)환경하에서의 폐기된 인증서 목록의 관리나 인터넷상에서 제3자가 발표한 정보의 무결성을 입증하는데도 응용될 수 있다.

1. 서 론

최근 무선인터넷 보급확대와 전자상거래 증가, 이동성 확대, 인터넷 사용 인구 증가 등으로 핸드헬드PC시장이 급속도로 증가하고 있으며 PDA를 이용한 증권거래 시스템과 이동통신회사들의 휴대전화를 통한 전자상거래 결제시스템이 서비스를 시작함으로써 소형기기를 이용한 전자상거래 및 전자금융거래가 활성화되고 있다. 그러나 현재 사용되고 있는 소형기기들이 아직까지는 전자상거래나 전자금융거래에 필요한 컴퓨팅 능력을 충분히 갖고 있지 못하며, 또한 통신네트워크상의 데이터 전송능력이 모자라고 불안정한 것이 현실이다. 따라서 이러한 소형기기들의 빠르고 안전한 전자상거래 및 전자금융거래를 위해서는 첫 번째로 소형기기들의 계산량을 줄여야 한다. 계산능력이 월등이 좋은 서버가 대부분의 계산을 하도록 한다던지 소형기기들이 해야할 계산을 서버가 미리 계산함으로서 소형기기의 계산량을 최대한 줄여야 한다. 두 번째로 소형기기와의 데이터 전송량을 줄여야 한다. 무선인터넷이라든지 모뎀을 이용한 통신과 같은 데이터 전송능력이 부족한 상황에서는 데이터의 손실이나 위변조등이 쉽게 일어날 수 있으며 이는 실질적으로 돈이 오가는 경제활동에서는 상당히 위험한 결과를 초래하게 된다. 마지막으로 소형기기를 포함한 모든 전자상거래나 전자금융거래를 이용하는 서비스 사용자가 늘어날 경우 발생하는 네트워크 과부하를 막아야 한다. 이러한 경우에 서버는 자신이 제공해야하는 정보를 여러개의 미러사이트를 통해 제공함으로써 사용자에게 좀더 빠르고 안정된 서비스를 제공할 수 있다. 또한 이러한 서비스의 분산으로 서버는 서비스거부(DoS)공격으로부터 보호 받을 수 있으며 미러사이트들 사이의 부하를 조절하여 전체적인 성능향상을 이루어 낼 수 있다. 그러나 이러한 미러사이트들의 무분별한 증가로 인한 정보의 무결성이 불투명해지는 문제가 발생할 수도 있다. 예를 들어 전자상거래를 위한 인증정보 또는 주식거래에 필요한 주식의 가격정보등이 미러사이트들의 조작으로 위변조될 경우 사용자는 큰 피해를 입게된다. 따라서 미러사이트들은 자신이 제공한 정보가 정보제공자로부터 받은 정보와 동일한 정보라는 것을 사용자에게 암호학적으로 검증해야한다.

위에 명시된 내용은 일반적으로 믿을 수 있는 정보제공자, 믿을수 없는 디렉토리 그리고 사용자로 표현해 볼 수 있다. 믿을 수 있는 정보제공자는 유한의 원소를 갖는 집합 S를 정의하고 각각의 원소들은 추가하거나 삭제할수 있다. 믿을수 없는 디렉토리는 정보제공자로부터 타임스탬프된 업데이트정보를 받아 집합 S의 복사본을 유지한다. 사용자는 질의를 통해 집합 S에 속한 특정원소의 존재여부를 묻고 디렉토리는 이에 대해 정보제공자가 서명한 인증정보를 포함한 예/아니오로 응답한다. 사용자는 디렉토리로부터 받은 응답메세지에서 정보제공자의 서명을 확인하여 정보의 무결성을 검증한다. 정보제공자와 디렉토리가 집합 S에 대해 질의와 업데이트 프로토콜을 유지하는데 사용하는 자료구조를 인증된 딕셔너리[2]라고 부른다. 아래의 [그림 1]은 인증된 딕셔너리의 구성도이다.



[그림 1] 인증된 딕셔너리

2. 관련 연구

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

2.1 일방향 어큐뮬레이터

일방향 어큐뮬레이터[1,3,4,5]는 제안하는 방법에서 가장 중요한 도구이다. 이 방법의 가장 일반적인 형식은 초기값 y_0를 시작으로 원소들의 집합 $X=\{x_1, x_2, ..., x_n\}$ 에 대해 일방향 함수 f를 이용하여 $y_i=f(y_{i-1}, x_i)$ 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.

2.2 오일러 이론

위에서 설명한 지수승 어큐뮬레이터에서 초기값으로 사용되는 y_0를 a라 하였을 때 p, q값과 서로소이며 역지수의 밑이되는 a값을 고르는 방법이 오일러 이론에서 명시된다.

오일러 이론 : 만약 a>1와 N>1이 서로소이면, $a^{\phi(N)} \bmod N = 1$

추 론 : 만약 a>1와 N>1이 서로소이면, 모든 x≥0에 대해서

$$a^x \bmod N = a^{x \bmod \phi(N)} \bmod N$$

이러한 추론은 인증된 딕셔너리에서 정보제공자만이 p, q값을 알고 있으므로서 디렉토리가 $\phi(N)$값을 계산하지 못하게 하여 계산된값에 대한 무결성을 유지할 수 있게 해준다.

3. 기존 연구

3.1 스트레이트포워드 방법

스트레이트 포워드 방법[6]은 집합 $S=\{x_1, x_2, ..., x_n\}$ 를 정보제공자가 가지고 있는 정보들의 집합이라 하고, 정보제공자는 서로 다른 큰소수 $p, q, N(=pq)$, 그리고 서로소인 밑수 a를 고른다. 또한 $A = a^{x_1 x_2 \cdots x_n} \bmod N$ 과 타임스탬프 t를 묶어 메시지 (A,t)를 전체 디렉토리에 전송한다. 이러한 초기화 과정이 끝나고 사용자가 원소 x_i가 집합 S에 속했는지에 대한 질의를 디렉토리에게 보내게 되면 디렉토리는 $A_i = a^{x_1 x_2 \cdots x_{i-1} x_{i+1} \cdots x_n} \bmod N$ 값을 계산하여 $A_i, N, (A,t)$를 사용자에게 되돌려 준다. 사용자는 디렉토리로부터 받은 메시지중 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 추가되는경우는 기존의 A에 추가되는정보 x_j승을 하여 새로운 A를 생성하고 삭제되는 경우는 해당되는 정보를 빼고 다시 A를 생성한다.

3.2 전처리 어큐뮬레이터

위의 스트레이트포워드 방법은 디렉토리가 질의를 받아서 해당되는 A_i를 만들어야하기 때문에 질의에 대한 응답시간이 늦어진다. 그러나 전처리 어큐뮬레이터[6]는 전 이진트리를 이용하여 집합 S에 속한 모든 정보들에 대하여 그들의 A_i값을 미리 계산해 놓음으로써 질의가 왔을 때 미리 계산된 A_i값을 전송하여 질의에 대한 응답시간을 단축시켰다. 이 방법은 다음의 두단계로 나누어 수행된다.

① 첫 번째 단계

전 이진트리 T를 구성하여 그 트리의 단말노드들에 집합 S에 속한 원소들을 배치한다. 그리고 트리 T를 후위순회로 운행하면서 각각의 노드 v에 대해서 $x(v)$를 계산한다. 만약 노드 v가 단말 노드일 경우 $x(v) = x_i \bmod \Phi(N)$ 로 계산되어지고 v가 내부노드일 경우는 v의 왼쪽 자식노드 u와 오른쪽 자식노드 w에 대해 $x(v) = x(u)x(w) \bmod \Phi(N)$ 를 계산하여 트리 T의 모든 노드들을 계산한다.

② 두 번째 단계

첫 번째 단계에서 계산된 전 이진트리 T를 이용하여 T의 루트노드 r에 대해서 $A(r)$=1로 정의한 후 전위순회로 트리 T를 운행하며 루트노드 r을 제외한 모든 노드들 v에 대하여 $A(v) = A(z)^{x(w)} \bmod N$ 을 계산한다. 이렇게 계산된 트리 T의 단말노드들에는 결국 각각의 원소들에 해당하는 A_i값들이 놓여지게 되고 사용자의 질의에 미리 계산된 A_i값을 보낼수 있게 된다.

이러한 두 개의 단계는 모두 정보제공자가 직접 계산하여야 하고 디렉토리는 정보제공자로부터 계산되어진 A_i값들만 갖게 되며 모든 업데이트 또한 정보제공자가 계산하여 디렉토리에게 전송한다.

3.3 파라미터를 이용한 어큐뮬레이터

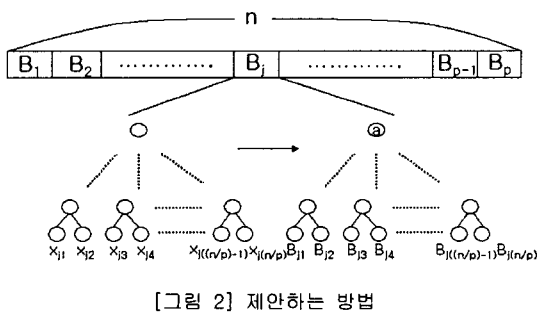
이 방법은 파리미터값 p를 이용하여 집합 S를 p개의 그룹으로 나누어 계산하는 방법[6]으로 p값은 $1 \le p \le n$의 범위 내에서 정보제공자와 디렉토리의 계산 능력에 따라 유동적으로 조절될 수 있다는 것이 장점이다. 우선 집합 S를 총 p개의 부분집합 $Y_1, Y_2, ..., Y_p$ 로 나누고 각각의 집합 Y_j에 속한 원소인 x_i들은 2진 탐색 트리로 구성되며 B_j는 x_i를 포함하는 Y_j를 제외한 나머지 집합들의 루트값들을 지수승하여 $B_j = a^{y_1 y_2 \cdots y_{j-1} y_{j+1} \cdots y_p} \bmod N$ 과 같이 계산하고 x_i에 해당하는 A_i값은 x_i가 속해있는 Y_j가 포함하는 범위를 $k \sim l$ 이라 하면 $A_i = B_j^{x_k x_{k+1} \cdots x_{j-1} x_{j+1} \cdots x_l} \bmod N$ 와 같이 계산할 수 있다. 이때 집합 Y_j의 원소들은 2진 탐색 트리로 저장되어 앞서 설명한 전처리 어큐뮬레이터의 첫 번째 단계와 같은 트리 연산을 통해 각각의 Y_j마다 개별적으로 트리를 구성하게 된다. 이러한 초기과정을 거쳐 사용자가 원소 x_i에 대해 질의를 보내게 되면 디렉토리는 x_i를 포함하는 Y_j의 트리에서 B_j를 계산하고 이를 이용해 A_i를 계산하여 기존의 방법과 같이 $A_i, N, (A,t)$를 사용자에게 되돌려 주게 된다. 또한 새로운 원소의 업데이트에 있어서 추가되거나 삭제되는 원소가 속한 부분집합 Y_j의 트리를 이용하여 B_j를 계산하고 총 p개의 부분집합들의 지수승을 통하여 A_i값을 계산하게 되므로 전처리 어큐뮬레이터에 비해 업데이트 시간을 단축시킬수 있다.

4. 제안하는 방법

제안하는 방법은 기본적으로 미리 계산된 증거값을 이용하여 질의에 대한 응답시간을 최소하는데 그 목적이 있으며 파라미터를 이용한 어큐뮬레이터와 같이 파라미터 p를 이용하여 새로운 원소의 추가와 삭제에 소요되는 시간을 단축하고자 하였다.

4.1 초기화

우선 전체 n개의 원소를 갖는 집합 S를 p개의 부분집합 $B_1, B_2, ..., B_p$ 로 나눈다. 다음으로 n/p개의 원소를 갖는 각각의 부분집합 B_j들을 앞서 말한 전처리 어큐뮬레이터와 같이 단말노드들에 부분집합의 원소들을 배열하고 단말노드일 경우 $x(v) = x_i \bmod \Phi(N)$, 내부노드일 경우 $x(v) = x(u)x(w) \bmod \Phi(N)$ 를 계산하여 트리 t를 구성하고 이 트리 t를 이용하여 $B(v) = B(z)^{x(w)} \bmod N$ 을 계산한다. 이렇게 되면 총 p개의 부분집합을 p개의 트리로 저장하게 된다. 아래 [그림 2]는 제안하는 방법의 구성도이다.



[그림 2] 제안하는 방법

이렇게 계산되어진 B_{ji}들과 증거 $(B_1,t), (B_2,t), ..., (B_p,t)$ 값들을 디렉토리에 전송하고 디렉토리는 정보제공자가 모두 계산하여 보내준 값을 저장한다.

4.2 질의 및 확인

사용자는 본인이 갖고 있는 정보가 집합 S에 속하는지를 확인하기 위하여 디렉토리에게 원소 x_i대한 질의를 보내게 되면 디렉토리는 x_i를 포함하는 (B_j,t) 값과 미리 계산되어진 B_{ji}값을 해당 사용자에게 보내주게 된다. 사용자는 B_{ji}값에 x_i승을 하여 B_j와 일치하는지 확인하게 된다. 이러한 방법은 p개의 부분집합에 속한 각각의 원소들에 대한 계산값이 미리 계산되어 있기 때문에 기존의 파라미터를 이용한 어큐뮬레이터 방법에 비해 매우 빠르게 질의에 대한 응답할 수 있게 해준다. 만약 이러한 인증된 딕셔너리가 PKI(Public Key Infrastructure) 환경에서 CRL(Certificate Revocation List)을 확인하는 방법에 적용되어 쓰여진다면 사용자가 질의를 보내 x_i가 집합 S에 속하는지, 다시말해 폐기된 인증서인지를 확인하는 과정에서 x_i가 집합 S에 속해있지 않을 경우도 생각해 볼 수 있다. 이러한 경우에는 특정원소의 계산값 B_{ji}를 보내줄 수 없으므로 Kocher [7]가 제안한 방법을 이용하여 각 원소들간의 범위를 나타내는 r_i=[x_i, x_{i+1}] 값으로 해쉬트리를 구성하고 이 트리의 루트값에 서명하여 질의에 응답해 줌으로서 사용자가 질의로 보낸 원소 x_i가 집합 S에 속해 있지 않음을 확인시켜줄 수 있다.

4.3 업데이트

우선 새로운 원소가 추가될 경우를 살펴보면 정보제공자는 자신이 갖고 있는 p개의 부분집합중에서 원소의 개수가 가장 작은 부분집합을 선택하여 해당되는 부분집합의 트리만을 추가된 원소와 함께 재구성하면 된다. 다음으로 원소가 삭제될 경우에는 삭제될 원소를 포함하고 있는 부분집합에서 해당원소를 삭제한후 역시 그 부분집합의 트리만을 재구성하면 된다. 이는 기존의 전처리 어큐뮬레이터가 전체 원소를 포함하는 트리를 재구성하는데 비해 매우 적은 계산량을 요구하게 된다.

4.4 성능분석

기존에 어큐뮬레이터를 이용한 인증된 딕셔너리는 앞서 2절 기존연구에서 밝힌바와 같이 세가지로 나누어 볼 수 있다. 이 방법들은 모두 사용자에게 적은 계산량을 할당하도록 설계되었으나 질의에 대한 응답이나 업데이트에 필요한 계산량이 상대적으로 많아지는 단점이 있다. 제안하는 새로운 방법은 이러한 단점들을 좀더 효율적으로 개선하고자한 것으로 아래 [표 1]은 기존의 방식들과 제안하는 방식의 실행시간을 비교한 표이다.

방 법	공간	삽 입	삭 제	업데이트 정보	질 의	질의 정보	확인
스트레이트 포워드	$O(n)$	$O(1)$	$O(n)$	$O(1)$	$O(n)$	$O(1)$	$O(1)$
전처리 어큐뮬레이터	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(1)$
파라미터를 이용한 어큐뮬레이터	$O(n)$	$O(p+\log(n/p))$	$O(p+\log(n/p))$	$O(p)$	$O(n/p)$	$O(1)$	$O(1)$
제안하는 방법	$O(n)$	$O(n/p)$	$O(n/p)$	$O(n/p)$	$O(1)$	$O(1)$	$O(1)$

[표 1] 실행시간 비교

[표 1]에서 보는 바와 같이 질의에 대한 실행시간은 $O(1)$로 파라미터를 이용한 어큐뮬레이터의 $O(n/p)$ 와 비교하였을 때 빠르게 실행되는것을 확인할수 있다. 이는 제안하는 방법이 열악한 네트워크 환경에서도 질의에 대한 빠른 응답이 가능하다는 것을 말하고 있다. 또한 삽입, 삭제와 같은 업데이트 시간은 파라미터를 이용한 어큐뮬레이터가 $O(p+\log(n/p))$ 이고 제안하는 방법이 $O(n/p)$ 으로 p값의 변화에 따라 유동적이다. 아래 [표 2]는 파라미터를 이용한 어큐뮬레이터와 제안하는 방법에서의 업데이트 시간과 질의에 대한 응답시간을 p값의 변화에 따라 비교한 것이다.

	업 데 이 트			질 의		
	$p=\sqrt{n}$	$p\rightarrow 1$	$p\rightarrow n$	$p=\sqrt{n}$	$p\rightarrow 1$	$p\rightarrow n$
파라미터를 이용한 어큐뮬레이터	$O(\sqrt{n}+\log n)$	$O(\log n)$	$O(n)$	$O(\sqrt{n})$	$O(n)$	$O(1)$
제안하는 방법	$O(\sqrt{n})$	$O(n)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$

[표 2] p값에 따른 비교

[표 2]에서는 p값이 $\sqrt{n}$을 기준으로 $p\geq\sqrt{n}$일 경우 파라미터를 이용한 어큐뮬레이터보다 제안하는 방법이 질의 시간과 업데이트 시간이 모두 월등히 빨라짐을 볼수 있다. 또한 $p\leq\sqrt{n}$일지라도 질의 시간이 파라미터를 이용한 어큐뮬레이터보다 빠르기 때문에 제안하는 방법이 열악한 네트워크 환경을 갖는 소형기기에서는 더욱 안정적으로 동작할 수 있다.

5. 결론 및 향후 연구

제안하는 방법이 기존의 방법들과 비슷한 업데이트 시간을 갖는 반면에 질의에 대한 응답 시간을 빠르게 함으로써 열악한 네트워크 환경과 제한된 계산능력을 갖는 소형기기들이 실제 전자상거래나 전자금융거래를 이용하는데 매우 유용하게 쓰일 수 있음을 보였다. 향후 제안한 방법을 기반으로 업데이트 시간 향상과 같은 개선점을 찾아 발전시켜 나가고, 기존방법과 제안한 방법에 대한 시뮬레이션을 통해 실질적인 검증을 해야할 것이다.

참고문헌

[1] 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.

[2] 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.

[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.