The Journal of Korean
Institute of CALS/EC
Vol. 7, No. 2, August 2002

H-이진트리체제의 타임스탬프 프로토콜 분석

정경원*, 도경화**, 전문석**

The Design & Analysis of Time-Stamp Protocol with H-Binary Tree

Kyung-won Jung, Kyoung-Hwa Do, Moon-Seog Jun

Abstract

We want to find a timestamping method which improves efficient performance and have high-level security to send secured messages in the digital signature and the law of e-commerces. Our paper shows a H-binary tree of time stamp to use a time stamp protocol with high security and performance in the packets of sending messages. We implement and analyze the protocols, show to compare with previous RSA methods. Our proposed protocol has O(log n) time complexity and high-performance.

Key word *: Time Stamp Protocol, Time stamp, E-Document, PKI*

* 세민정보산업고등학교
** 숭실대학교 컴퓨터학과

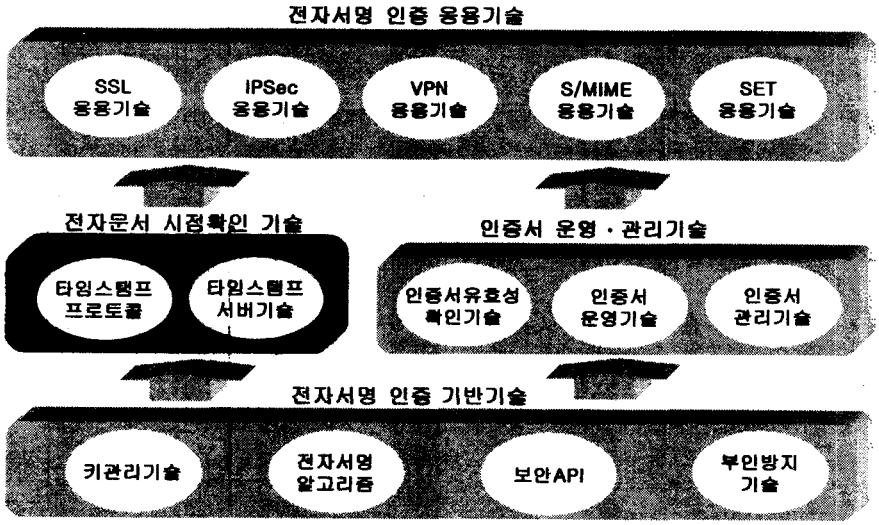
1. 서 론

지난 몇 년 동안 전자정부 구현을 위하여, 전자서명을 이용한 상거래를 합법적으로 규정하고 시행 중에 있다. 그러나, 타임스탬프의 중요성에 비하여 아직 발전되지 않고 있는 실정이나 타임스탬프의 이론적이고, 합법적인 측면이 세계적인 관심의 주제가 되고 있다.

타임스탬프(Time-Stamp)는 전자상거래에서 안전한 전자문서를 생성하거나 폐기하고자할 경우, 발행시점의 시간을 서명함으로서 상호인증관계의 시점을 확인해 주는 기법이다. 타임스탬프의 실제 중요성은 전자문서의 합법적 사용을 위해 유효기간을 확인할 필요가 있을 때, 이를 명백하게 하는 것에 있다.

<그림 1>은 전자서명 인증기반 기술에서 작성된 내용을 중심으로 전송하려는 모든 전자문서를 시점 확인하는 기술이 PKI에서 현재 중요한 기술임을 보여주고 있다. 타임스탬프의 안전성과 비도 향상을 위한 방법은 세계적으로 계속 발전되어 왔지만 국내에서는 아직 활발하게 전개되지 못하고 있다. 그러나 본 기술은 전자상거래 SSL 응용기술, IPSec 기술, VPN, S/MIME 등에 응용되어 보안산업에 중요한 기술로서 평가받고 있다.

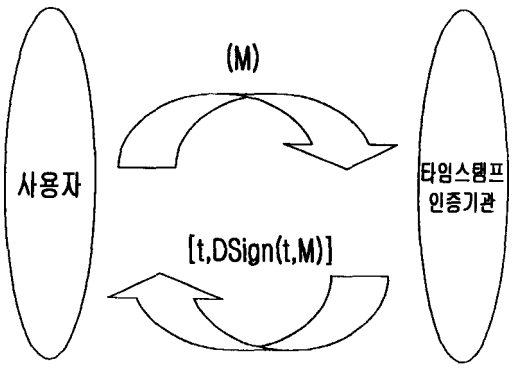
기존 타임스탬프 시스템[1, 2, 6, 7, 9, 11, 13, 14]은 타임스탬프의 비선형부분 정렬을 사용하는데, 이는 해커들에게 취약점을 제공한다. 본 논문에서는 이론을 중심으로 제안된 타임스탬프 시스템의 프로토콜을 설계하고, 실제적인 구현을 통해서 분석하였다. 타임스탬프 없는 원시적인 암호 방법으로 서명된 문서는 신뢰할 수 없게 되거나, 서명자 스스로 서명을 거절하는 경우의 문서도 신뢰할 수 없게 된다. 이러한 문제점들을 해결하고, TCP/IP, OSI, 윈도우 환경에서 쉽고, 비도가 높은 타임스탬프를 제공하는 방법을 찾는데 본 논문의 목적이 있다.



<그림 1> PKI 상호연동을 위한 상호인증기술

2. 기존 타임스탬프 프로토콜

간단한 타임스탬프 프로토콜[2, 11, 13]의 개념은 송신자의 보내려는 전자문서(M)에 현재시간을 추가하여 수신자에게 전송함으로서, 전자상거래에서 발생되는 전자문서의 안전성에 대한 법적 근거를 얻기 위하여 시간을 추가, 인증하는 시스템 프로토콜을 말한다. 타임스탬프 시스템(TSS)의 수행 과정은 해쉬함수를 이용하여 전자문서 M에 현재시간 t와 합성문서 (t, M)의 전자서명값 $DSign = DSign(t, M)$를 수신자에게 보내는 시스템을 말한다.



<그림 2 기본 타임스탬프 구조

현재 사용하고 있는 GPS를 이용한 시간 t가 주어졌다고 가정하고 이론적인 면에서 본 논문을 새로운 H-이진트리 타임스탬프 프로토콜을 설계한다.

기존의 타임스탬프 시스템[4, 5]의 단점은 간단하고, 비도가 낮은 비현실적인 값을 전송하는 구조를 갖고 있다. 즉, 모든 전송과정을 무조건 신뢰하고, 해커로부터 공격이 없는 조건상에서 순수하게 타임스탬프 시스템의 현재 시간 t에 발행된 디지털 서명값 $DSign$이 검증되어진 값인지, 아닌지가 불가능한 구조에 있다고 볼 수 있다. 이러한 문제점을 해결한 더욱 안전한 타임스탬프 시스템은 해쉬함수 사용과 비밀키와 공개키 사용 문제에 대하여 그림을 통해서 쉽게 해결할 수 있다.

<그림 3>은 일반적인 전자서명이 단방향 해쉬함수를 이용하여 전자문서를 전송하는 구조를 보여주고 있다. 수신된 문서에서 수행된 전자서명값과 전자문서에 포함된 전자서명값을 비교하여 신원확인과 문서 확인을 통해서 프로토콜을 검증하는 과정을 설명하고 있다.

최적해를 얻기 위해서 Prefix 이진트리 구조의 타임스탬프 시스템을 이용하여 타임스탬프 서명을 수행할 수 있다. 해쉬함수 H는 일반적으로 MD5와 SHA-1를 많이 사용하고 있으며, 이론적인 수식으로 수행과정을 표현할 수 있다.

n-번째의 송신자의 전자문서 M_n에 대한 타임스탬프 전자서명은

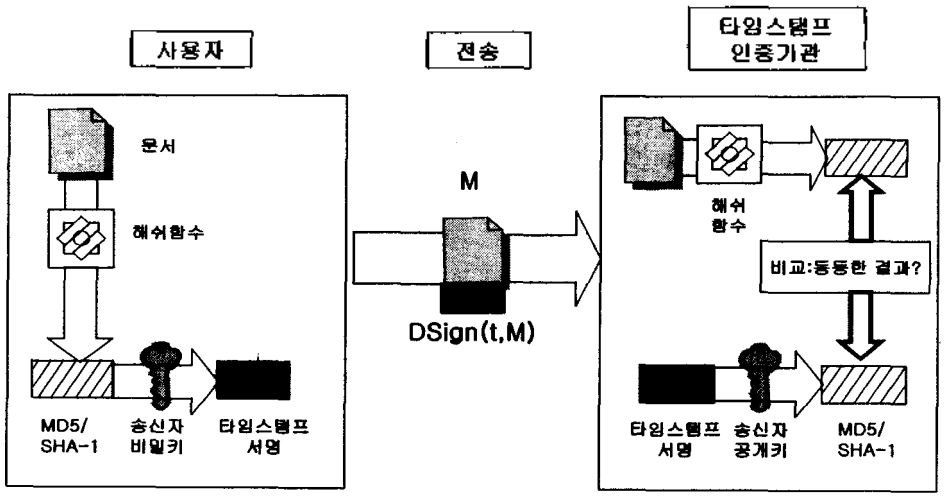
$DSign = DSign(n,\ t_n,\ CertID_n,\ M_n,\ P_n)$

여기서, t_n은 현재의 시간, $CertID_n$는 사용자 식별자이고, C_n은 이진트리 구조에서 재귀함수 수식을 계산한 정보로서

$C_n := (t_{n-1},\ ID_{n-1},\ M_{n-1},\ Hash(C_{n-1}))$

으로 정의된다.

이러한 전자서명 체계는 전자문서만 안전하게 송신할 수 있는 구조를 갖고 있지만, 다양한 전자문서가 여러 계층을 통해서 전자상거래를 하는 경우 더욱 강력한 전자서명과



<그림 3> 일반적인 전자서명 방법

타임스탬프가 요구된다. 전자서명 방법으로 MD5를 사용하였으며, H-이진트리 체계에서 반복적으로 X.509 형태의 타임스탬프가 구현된다. 이러한 구조는 본 논문을 통해서 안전하고, 간단한 타임스탬프임이 이론적인 면에서 증명된다. 역함수를 통해서 암호화된 문서가 정확하게 원래의 문서로 환원된다.

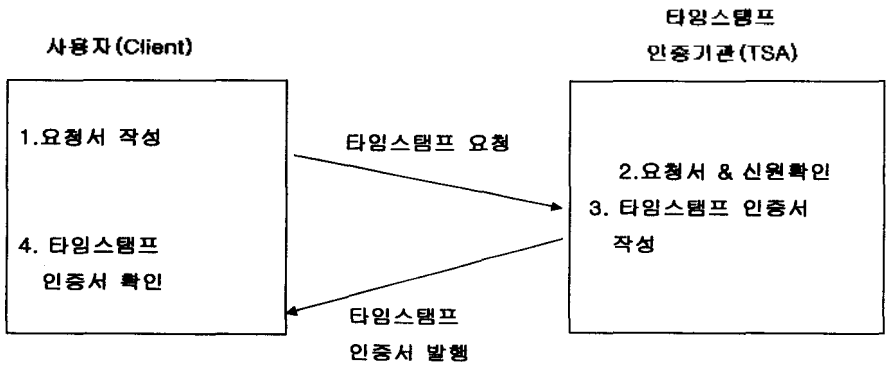
3. 타임스탬프 프로토콜 설계

3.1 기본 프로토콜 구조

<그림 4>는 안전한 전자상거래에서 전자문서에 시점확인 인증서를 포함하여 전송하기 위하여, 가장 기본적인 구조에서부터 발전된 타임스탬프 프로토콜을 설계하였다. 수행하는 도중 해커로부터 해킹방지를 위한 방법으로 타임스탬프의 비도를 높게 하고자 한다. 이러한 경우, 라운드를 반복해서 비도를 높일 수 있다. 다음과 같은 기본단계의 수행알고리즘을 통해서 프로토콜을 설계할 수 있다.

1. 사용자는 타임스탬프 인증프로그램을 설치한 후에 타임스탬프를 요청할 전자문서를 작성하고, 타임스탬프를 타임스탬프 인증기관(TSA)에 요청한다.
2. 타임스탬프 인증기관(TSA)은 요청서를 확인하고, 사용자의 신원을 확인후, LOG에 기록한다.
3. TST(Time-Stamp Token)를 생성하여, 타임스탬프 인증기관의 서명키로 서명한 후, 타임스탬프 인증서를 작성하여, 사용자에게 인증서를 발행한다.
4. 사용자는 TST의 내용 및 타임스탬프 인증기관의 서명을 자신이 갖고 있던 타임스탬프 인증기관의 인증서로 확인후, TST와 요청한 문서의 정보를 묶어서 보관한다.

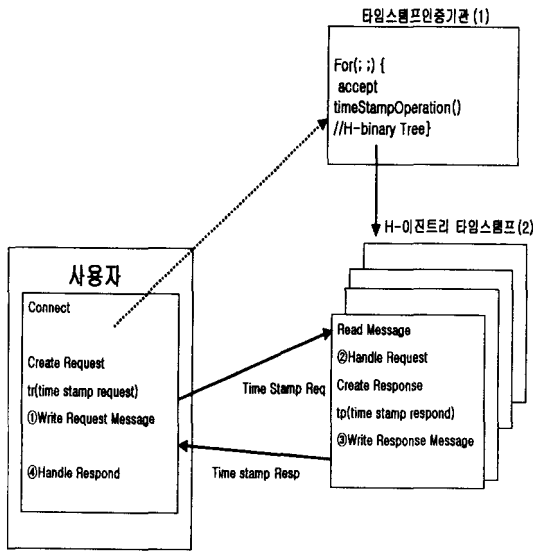
타임스탬프 기본 프로토콜은 다음 구현 프로그램 구조형태로 변환 가능하며, 각각의



<그림 4> 기본 타임스탬프 프로토콜 구조

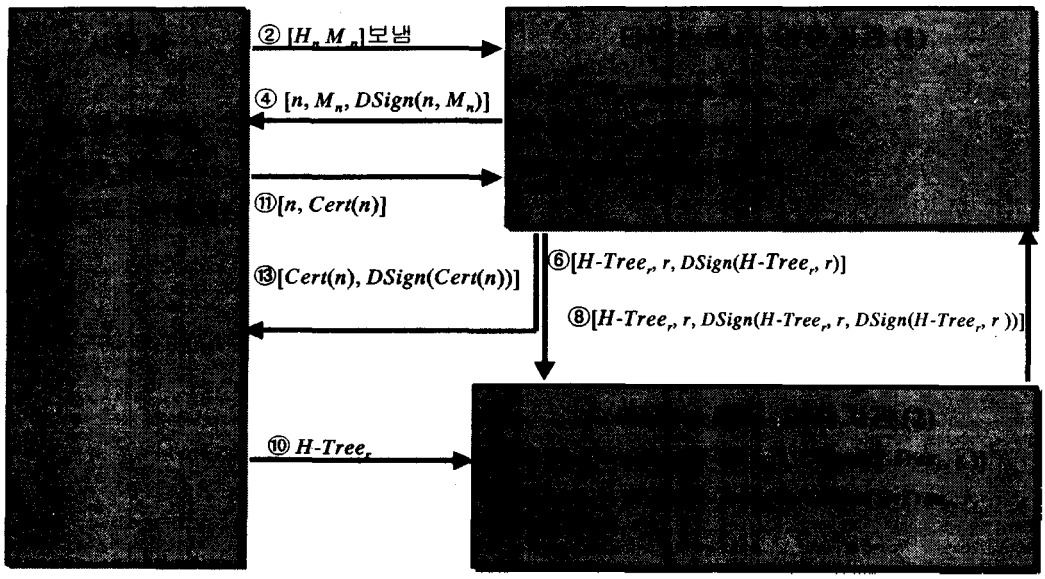
기능은 함수프로그램으로 타임스탬프 인증기능을 수행한다. 타임스탬프 요청서는 타임스탬프 인증등록기관에 보내는 TimeStapReq를 이용하여 요청할 수 있다. 요청서에 대한 응답서는 TimeStapResp를 통하여 응답하며, 이는 타임스탬프 토큰에 포함된 전자서명과 신원확인 내용을 사용자에게 전달한다.

<그림 5>는 구현하기 위하여, 간단하게 설계한 구조이다. 제안된 H-이진트리 타임스탬프 구조는 시점확인면에서 안전하고, 정확한 정보보호에 중요한 프로토콜로서 구현될 것이다. 통신 프로토콜 구조는 3-way handshake 형태에 의해서 요청서를 먼저 보내고, 상대방으로부터 연결허락 Ack를 받으므로서 연결이 가능하게 된다. 이러한 프로토콜 구현을 위해서 타임스탬프 요청서를 사용자가 해쉬함수를 이용하여 생성하며, 타임스탬프 라운드 버전이 포함되며, 타임스탬프 인증기관에 요청하게 된다. 인증기관은 사용자의 신원을 확인한 후에 사용자와 연결이 가능하게 된다. 연결이 허락된 후에 실제 타임스탬프 요청서 TimeStampReq를 보내고, 응답서 TimeStapResp를 사용자가 받게 된다. 간단한 구조는 구현형태로서 프로토콜를 작성할 수 있다.



<그림 5> 타임스탬프 구조 구현

1. 요청서 작성: Write tsaMsg
2. 요청서 및 신원확인: Handle request
3. 타임스탬프 인증서 작성: Write Respond Message
4. 타임스탬프 인증서 확인: Handle respond



<그림 6> 제안된 타임스탬프 프로토콜

3.2 제안된 타임스탬프 프로토콜

<그림 6>은 제안된 타임스탬프 프로토콜 구조로서 인증기관과 사용자 사이에서 송수신하는 전자서명과 신원확인에 필요한 기본 구조로 작성되었다. 기본 준비단계로서 사용자는 인증기관으로부터 사전에 인증프로그램을 다운로드하고, 사용자는 발행기관으로부터 인증서를 받은 상태에서 다음과 같은 수순에 의해서 타임스탬프 프로토콜을 수행한다.

(단계 1) 타임스탬프 발행기관에서 타임스탬프 프로그램 설치 후, 해쉬함수 H_n생성.

(단계 2) 전자문서 M_n의 타임스탬프 발행을 위하여 요청서 [H_n, M_n]를 보내며, 해쉬함수를 이용해서 상호신원확인을 위한 요청서임.

(단계 3) 발행기관 TSA에서 사용자로부터 받은 요청서 [H_n, M_n]를 갖고, 사용자의 신원을 확인하며, 전자서명 $DSign(n, M_n)$를 작성. 사용자 신원확인:

$$[n, M_n, DSign(n, M_n)]$$

(단계 4) 상호인증과정으로 전자서명을 사용자에게 되돌려 보낸다. 발행기관신원확인:

$$[n, M_n, DSign(n, M_n)]$$

(단계 5) H-이진트리 타임스탬프의 현재 라운드와 내용을 전자서명 계산한 후, 타임스탬프 서버에 저장함.

$$[\ H-Tree_r, r,\ DSign(\ H-Tree_r, r)]$$

(단계 6) 타임스탬프 발행기관(2)에 현재의 라운드와 서명이 포함된 새로운 타임스탬프 요청서를 보낸다.

$$[\ H-Tree_r, r, DSign(\ H-Tree_r, r)]$$

(단계 7) 발행기관(1)의 신원확인과 상호인증과정으로 타임스탬프 서명을 생성한다.

$[\ H-Tree_r, r,\ DSign(\ H-Tree_r, r,\ DSign(\ H-Tree_r, r))]$

(단계 8) 타임스탬프 인증서와 전자서명내용을 발행기관(1)에 보낸다.

$[\ H-Tree_r, r,\ DSign(\ H-Tree_r, r,\ DSign(\ H-Tree_r, r))]$

(단계 9) 타임스탬프 인증서 발행에 대한 타임스탬프와 서명내용을 서버에 저장

$[\ H-Tree_r, r,\ DSign(\ H-Tree_r, r), DSign(\ H-Tree_r, r,\ DSign(\ H-Tree_r, r))]$

(단계 10) 사용자가 발행기관(2)에게 타임스탬프 $H-Tree$의 라운드를 확인한다.

(단계 11) 타임스탬프요청서 $[\ n,\ Cert(n)]$를 발행기관(1)에게 요청한다.

(단계 12) 타임스탬프와 서명내용$[\ Cert(n), DSign(Cert(n))]$을 생성한다.

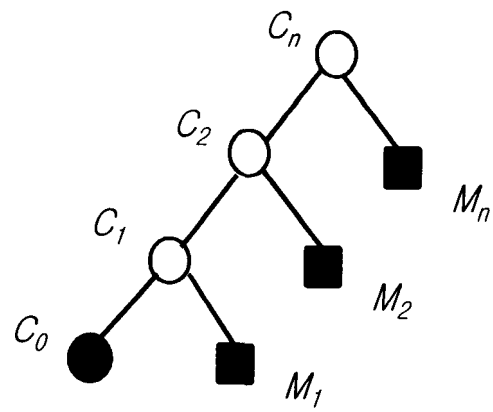
(단계 13) 타임스탬프 인증서$[\ Cert(n), DSign(Cert(n))]$를 사용자에게 최종적으로 보낸다.

타임스탬프 인증기관(1)은 PKI의 인증등록기관과 유사하며, 인증기관(2)는 인증기관과 같은 기능을 하며, X.509 구조에 발행된 타임스탬프를 발행함으로서 발행 된 시점확인 기능을 전자문서에 추가하는 알고리즘을 작성할 수 있다. $Cert(n)$는 인증된 내용을 의미하며, $DSign(n)$는 전자서명을 의미한다.

3.3 H-이진트리 타임스탬프 체계

일반적으로, 컴퓨터 알고리즘에서 최적해를 갖는 형태는 트리구조를 갖는 검색구조로서 O(log n)의 시간복잡도를 갖게된다. DES 알고리즘은 16번의 라운드를 통해서 얻어진 결과를 이용함으로서 비도를 높이고 있고, RSA 알고리즘도 높은 지수승을 이용하여 해커로부터의 복호화를 방지하고 있다. 본 논문에서는 반복되는 라운드의 결과를 이용하여 최적의 시간복잡도를 얻기 위해서 H-이진트리 구조를 이용한다. 커다란 전자문서 M를 송신자가 수신자에게 전송하려고 하는 경우, 전자문서는 같은 크기로 분할된 패킷 메시지 M_i들이 만들어진다.

$$M = \{\ M_i \mid 1 \le i \le n\ \}$$



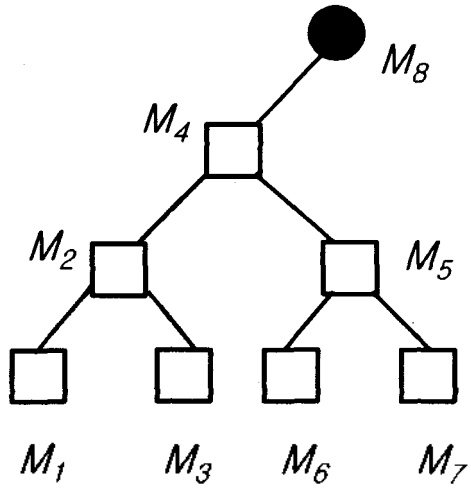
<그림 7> Prefix-이진트리

<그림 7>은 간단한 Prefix 이진트리구조이며, 해쉬함수 Hash를 이용하여 반복 라운드 i 만큼 비도를 향상시키기 위하여 다음과 같은 수식을 통해서 암호화된 패킷 C_i 얻을 수 있다.

$$C_1 = \text{Hash}(\ M_1,\ C_0\ )$$
$$C_2 = \text{Hash}(\ M_2,\ C_1\ )$$
$$C_3 = \text{Hash}(\ M_3,\ C_2\ )$$
$$\vdots$$
$$C_n = \text{Hash}(\ M_n,\ C_{n-1})$$

여기서, 전송하려는 전자문서(M_1, M_2, M_3, …, M_n)는 타임스탬프가 포함된 암호화된 문서(C_1, C_2, C_3, …, C_n)으로 변환되는 시스템을 구현할 수 있다. 이러한 구조는 문서형태, 그림, 동영상 등에 응용이 가능하며, 전자문서의 디지털 저작권 관리(DRM : Digital Right Managements)기능으로 발전되는 기본구조가 된다. 또한, 이러한 형태는 OSI, TCP/IP, ATM 구조에서 계층과 계층 사이에서 송신자측에서 암호화구조와 수신자측에서 복호화 구조 형태와 유사한 구조이다. 특이하게 다른 점은 타임스탬프를 추가하여 발행된 시점을 패킷에 추가하였다는 점이 다르다.

<그림 8>은 H-이진트리 타임스탬프 체계를 작성한 구조로서, 비도를 향상하기 위한 방법은 다양한 분야에서 연구가 되고, 개발이 진행되고 있다. 본 논문은 수행속도가 빠르고, 최적해를 얻을 수 있는 제안된 H-이진트리구조를 이용하여 기본원리를 발전시키고자 한다. 최적화된 시간스탬프 인증서를 얻기 위한 방법으로 제안된 H-이진트리구조에서 검증하여 보자. 깊이 d인 시간 인증구조의 길이는 $k \cdot (d+1)$되며, 시간복잡도는 O(log n)이 된다.



<그림 8> H-이진트리 타임스탬프 체계

모든 패킷 메시지를 갖고 타임스탬프를 작성한다는 것은 시간복잡도가 증가하기 때문에 근처에 있는 패킷 메시지만을 갖고서 타임스탬프를 작성하는 방법을 응용하고자 한다. H-이진트리 체계에서 해쉬함수 C_n의 재귀적인 연산에서 타임스탬프 t_n이 수식에서 생략되어있지만, 실제로 전송하려는 전자문서 M_n의 현재 시점시간으로 표현되기 때문에, 암호화된 타임스탬프 내용 속에 포함되어있다.

$$C_1 = \text{Hash}(\ M_1,\ C_0\ ),$$
$$C_2 = \text{Hash}(\ M_2,\ C_1\ ),$$
$$C_3 = \text{Hash}(\ M_3,\ C_2\ ),$$
$$C_4 = \text{Hash}(\ M_4,\ C_2\ ),$$
$$C_5 = \text{Hash}(\ M_5,\ C_4\ ),$$
$$C_6 = \text{Hash}(\ M_6,\ C_5\ ),$$
$$C_7 = \text{Hash}(\ M_7,\ C_5\ ),$$

C_8 = Hash(M_8, C_4),

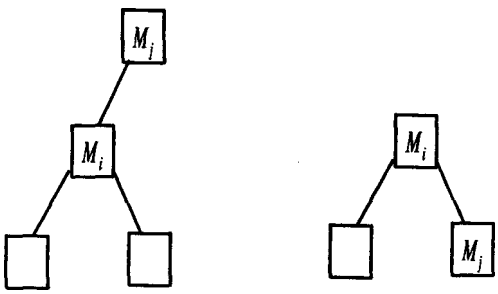
⋮

C_n = Hash(M_n, C_x),

x는 H-이진트리 구조상의 인덱스.

H-이진트리체계에서 M_i가 추가될 때마다, Hash함수를 이용하여 새로운 C_i를 생성한다. 인덱스 i는 H-이진트리체계에 추가되는 순서를 말하며, 분할된 전자문서의 순서를 의미한다.

(정리 1). H-이진트리에서 입력정점 M_j이 가장 가까운 이웃정점 M_i과 연결된 이진그래프를 형성할 경우, 시간복잡도 O(log n)와 정점 C_j = Hash(M_j, C_i)을 만족한다. (단, 입력순서: $M_1 \leq M_2 \leq \ldots \leq M_i \leq M_j$)



<그림 9> 입력 정점 M_j인 구조

Heap 이진트리구조의 형태에서 작성된 그래프로서, H-이진트리 구조에 대한 시간복잡도는 O(log n)임을 암호학 알고리즘[3]을 통해서 증명되었으므로, (정리1) 대한 증명은 생략한다.

<그림 9>는 한 정점을 입력하였을 경우, 작성되는 트리구조를 설명하고 있고, (정리 1)에서 증명에 필요한 구조이다. H-이진트리 타임스탬프 프로토콜은 단방향 해쉬함수에 의해서 생성되고, 전자서명을 작성하게 되므로, 정점 상호간의 충돌과 관계없이 O(log n) 속에서 안전한 타임스탬프를 생성하고, 역함수속에서 복호화 단계를 검증할 수 있다. H-이진트리체계는 재귀적인 공식을 만족하는 부분재귀함수 C_n를 함께 사용하며, 반복적인 암호화된 정보가 작성된다. 암호화된 정보는 Hash함수가 계산될 때마다, 타임스탬프가 정보에 추가되어진다.

$$C_n = Hash(M_n, Hash(M_{n-1}, Hash (M_{n-2}...(Hash(M_1, C_1)...))))$$

여기서,

타임스탬프 $C_n = Hash(M_n, C_x)$

의 내용에서, M_n는 n 번째 타임스탬프된 전자문서를 정의한다. (정리 2)는 (정리 1)의 역함수이며, 같은 시간복잡도를 갖고 검증한다. 역함수 $Hash^{-1}$는 전자서명 방법을 이용하며, 실제적으로 MD5 혹은 SHA-1를 이용하고 있다.

(정리 2). H-이진트리에서 출력정점 M_j에 가장 가까운 이웃정점 M_i과 연결된 이진그래프를 형성할 경우, 시간복잡도 O(log n)와 정점 $C_i = Hash^{-1}(M_j, C_j)$의 역함수가 성립한다. (단, 입력순서: $M_1 \leq M_2$

$\leq \ldots \leq M_i \leq M_j$)

$$C_1 = Hash^{-1}(M_1, Hash^{-1}(M_2, \ldots Hash^{-1}(M_3, \ldots, Hash^{-1}(M_n, C_n)\ldots))))$$

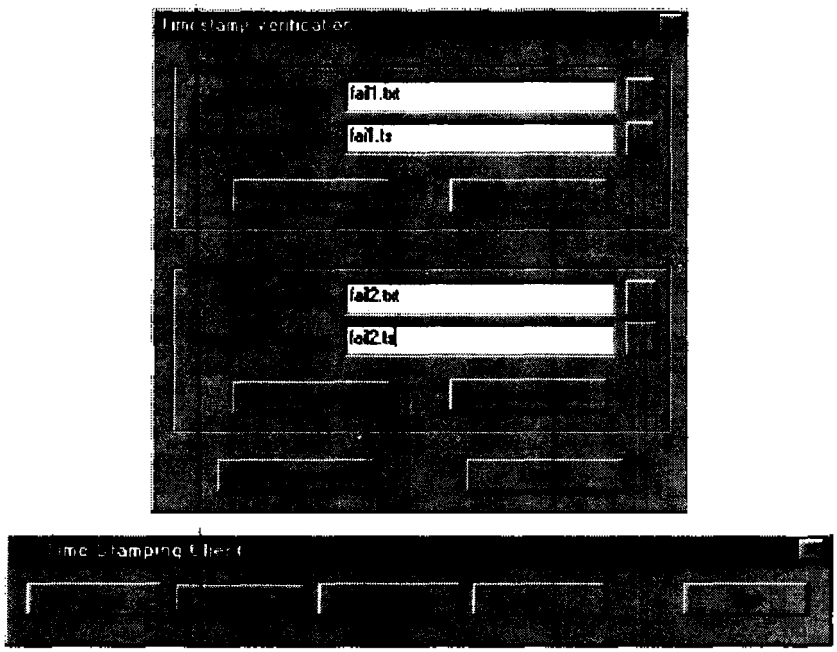
(정리 3). H-이진트리를 이용한 타임스탬프의 최대의 암호화 전자서명횟수는 $2(n-1)\lfloor \lg n \rfloor$ 를 수행하며, 그때, $H: N \rightarrow 2^N$는 이진함수관계를 갖으며, Heap-이진트리를 성립한다.

1. 임의의 두 개의 입력 전자문서 M_i과 M_j에 대하여, H-타임스탬프는 순서적으로 Heap 트리구조의 연결된 이진트리를 형성한다.
2. 모든 n에 대하여, Max {H(n)} - n ≤ c_2 (log n) = O(log n)성립한다.

4. H-이진트리 타임스탬프 구현 및 분석

H-이진트리 타임스탬프를 구현하기 위해서, RSA를 수정하고, 해쉬함수는 MD5를 이용하는 방법으로 다음과 같은 타임스탬프 인증서를 발행해서 사용자가 이용하게 된다. 실험환경은 다음과 같은 조건 하에서 다른 종류의 타임스탬프를 갖고 시험하였다.

다음과 같은 장비와 통신환경하에서 타임스탬프 시험을 통해서 구현하였다. <그림 10>은 타임스탬프 검증과 사용자 아이콘을



<그림 10> 타임스탬프 검증

작성하였으며, <그림 11>은 H-이진트리에서 발생된 타임스탬프 인증서 X.509 구조를 보여 주고 있다.

- ○ 시스템: Ultra-80 SUN
- ○ 시스템 소프트웨어: Solaris 2.8 & gcc
- ○ CPU: 400MHz

타임스탬프는 MD5 전자서명과 인증서를 보여주고 있다. H-이진트리 타임스탬프 구현과 분석을 통해서, 검증 프로토콜의 단일 실행 동안에 검사된 타임스탬프들의 수는 O(log n)임을 시험을 통해서 얻을 수 있다.

1) 정보보호를 위한 전자서명 알고리즘 MD5와 사용한 컴퓨터 환경은 동등한 구

```
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 0 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=EE, O=Cybernetica, CN=Pilot Timestamp Server
        Validity
            Not Before: Aug 22 02:11:54 2000 GMT
            Not After : Aug 21 02:11:54 2001 GMT
        Subject: C=EE, O=Cybernetica, CN=Pilot Timestamp Server
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:ca:44:5e:c6:73:ca:19:41:a7:ed:46:9d:89:4e:
                    1f:51:51:4b:92:e2:49:2a:72:3e:a9:70:cc:b7:cc:
                    5a:29:df:ec:74:11:fc:58:27:e9:ec:5e:89:d7:c2:
                    3c:83:ae:6e:79:e0:99:20:22:ef:b7:5b:50:8a:5c:
                    25:5a:8d:21:c6:58:c2:89:9c:18:f0:ff:78:5d:2d:
                    6b:53:2b:c8:6b:d9:2f:20:38:83:ff:90:1c:c6:d8:
                    84:3a:ef:53:95:65:e9:04:44:76:00:d6:7c:0b:13:
                    b7:5d:32:d8:58:87:28:c5:e9:5f:8f:eb:f1:de:73:
                    d7:0e:b1:12:7f:09:ec:95:73
                Exponent: 65537 (0x10001)
    Signature Algorithm: md5WithRSAEncryption
        37:31:cb:09:ae:52:9a:cc:0b:4f:42:d5:e5:ba:6b:83:85:44:
        aa:9a:47:27:e0:11:73:d5:9c:e1:1a:85:f7:46:36:31:d1:70:
        22:14:79:aa:76:42:4d:08:18:05:28:18:20:3d:d4:22:39:7c:
        35:45:af:11:76:00:69:2b:d7:46:e9:bb:8c:1d:21:fa:8b:4b:
        e5:15:96:3e:97:16:91:7d:e5:98:ef:ef:c5:11:21:2c:42:81:
        b6:bf:bd:44:27:84:09:f0:ef:3a:c2:17:52:7d:90:27:01:be:
        c7:e5:81:18:03:65:a7:52:15:a6:dc:5b:d9:0b:f6:5e:11:44:
        8e:14

-----BEGIN CERTIFICATE-----
MIIB9zCCAWACAQAwDQYJKoZIhvcNAQEEBQAwRDELMAkGA1UEBhMCRUUxFDASBgNV
BAoTC0N5YmVybmV0aWNhMRswGQYDVQQDExJQaWxvdCBUaW1lc3RhbXAgU2VydmVy
MB4XDTk5MDgyMjAyMTE1NFoXDTAwMDgyMTAyMTE1NFowRDELMAkGA1UEBhMCRUUx
FDASBgNVBAoTC0N5YmVybmV0aWNhMRswGQYDVQQDExJQaWxvdCBUaW1lc3RhbXAg
U2VydmVyMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDKRF7GcBoZQaftPp2J
Th9RUUsyokkqcj6ocMy3zFop3650EfxZJ6ms+pnXwIyTrm554JkgIu+3W1CKXCVa
jSH8MKJrBjw/3hdLMt TK8hr2S8g0IP/kBz62IQ6710VZekERHYA1nwLE7ddMtNW
hyjFqV+Pq/HecBc0sRJ/ieyVcwIDAQABMA0GCSqGSIb3DQEBBAUAA4GBADcxymu
UprMC09C1eW8e4OFRKqdR5fgEXPVn0Eahfd6NJ/RcCIUeTN2QkOIGAUo+CA91C15
fDVDrxF4AIkr1DjpU4wdIfprS+X1Ij6X9pF9pZjv78X/ISxCgba/vUQnhAnwrzrC
F1J9tCcBvsfIgAg0ZadSFabc+9kL9I7xRI4U
-----END CERTIFICATE-----
```

<그림 11> 타임스탬프 인증서 구조

조를 사용하고, 시험에 사용하는 암호키는 부분키 생성 알고리즘을 반복적으로 수행하여 이를 평균으로 산출한다.

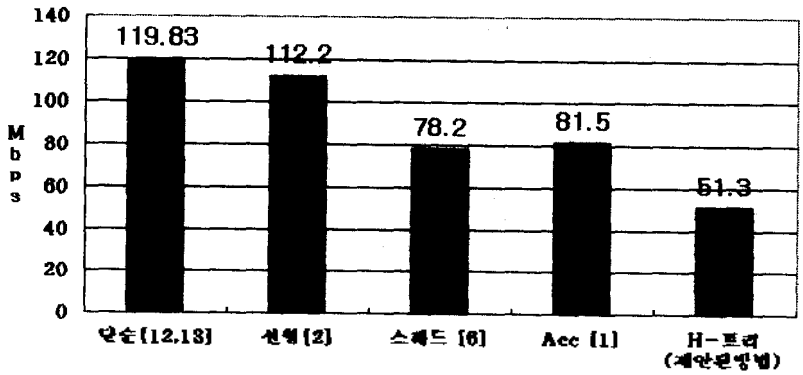
2) 해쉬함수는 MD5 알고리즘을 이용하였고, 임의의 단위 블록을 반복적으로 암호화하는 과정과 이것을 반복적으로 복호화 하는 과정을 수행하여 이를 평균으로 산출한다.
3) 각 타임스탬프 방법에 대하여 부분 키 생성 시간과 암/복호화 처리 시간을 공정하게 측정하기 위하여 모든 타임스탬프 방법을 동일하게 1분간 수행하여 그 속도를 측정한다.

<그림 12>의 결과는 제안한 H-이진트리 타임스탬프는 스레드방법과 비교해서 월등하게 속도면에서 우수하고, 시간복잡도면에서 빠른 결과를 얻었다. 가장 단순한 방법은 단순방법과 선형방법이지만, 전자서명을 반복적으로 수행함으로서 얻을 수 있는 타임스탬프는 H-이진트리임을 알 수 있다.

<도표 1>은 이론적인 면을 중심으로 시간복잡도를 비교분석 하였고, 다른 방법들의 특징과 문제점을 비교하였다. 간단하고, 전자서명을 반복적으로 수행하는데 필요한 타임스탬프 알고리즘은 H-이진트리가 우수함을 알 수 있다. 실제적인 구현을 통해서 각 방법에 대한 구현 속도는 국내에서 아직 확산되지 않은 타임스탬프 부문에 중요한 자료가 될 것이다.

<표 1> 타임스탬프 시간복잡도 & 보안평가

타임스탬프 방법	시간 복잡도	보안 평가	
		비도 등급	중요 특징과 문제점
(1) 단순타임스탬프 방법[12, 13]	O(n)	3	- 발행자와 공격자와의 충돌 가능성 - 발행자에 공격자의 개인적 개입 가능성
(2) 선형링크 방법[2]	O(n)	3	- 발행자와 공격자와의 충돌 가능성 - 발행자에 공격자의 개인적 개입 가능성
(3) 스레드인증 방법[6]	O(log²n)	6	- 디지털 서명 보안 강화 - 발행자와 공격자와의 충돌 가능성 - 발행자에 공격자의 개인적 개입 가능성
(4) Accumulated 타임스탬프[1]	O(log²n)	6	- 디지털 서명 보안: 높은비도 - 발행자와 공격자와의 충돌 가능성 - 계산양이 너무 많고 복잡
(5) H-이진트리 타임스탬프 [제안방법]	O(logn)	6	- 디지털 서명 보안: 높은비도 - 발행자와 공격자와의 충돌 없음 - 간단하고, 비도가 높은 방법



<그림 12> 타임스탬프 비교분석

5. 결 론

본 연구에서 구현된 H-이진 트리 타임스탬프는 X.509 인증서에 추가할 수 있으며, Notary 시스템을 구현해서 전자문서와 인터넷 서비스에 응용이 가능하다. 타임스탬프는 전자상거래에서 시각확인점을 얻고, 반복되는 암호화 작업을 통해서 인증서의 비도를 높인다는데 중요한 결과로 평가된다. 현재 국내의 몇몇 인증기관만이 인증서를 발급하고 있는 현실이지만, 민간 기관에서도 거래된 시각에 대한 타임스탬프를 발행해 주는 다양한 전자상거래 시각확인 인증기관이 설립되리라고 본다. 전자상거래 법에 의해서 거래된 전자문서의 시각을 추가함으로서 신원확인과 시각확인에 도움이 되리라고 본다.

현재 많이 이용하고 있는 ERP, DRM, Watermarking 기술에 응용이 가능하게 될 것이다. 본 연구에서 개발된 H-이진트리 타임스탬프는 효율적인 시간복잡도와 간단하게 전자상거래에서 응용이 가능하다는 면에서 다른 방법보다 우수하다고 판단된다. 국내에서 아직 연구가 미진한 시각확인 타임스탬프 연구가 더욱 발전되어야 할 것으로 판단된다.

참고문헌

[1] Ahto Buldas, Helger Lipmaa, Berry Schoenmakers, "Optimally Efficient Accountable Time-Stamping", PKC '2000, http://home.cyber.ee/helger /papers/bls00.html.

[2] Ahto Buldas, Peeter Laud, Helger Lipmaa, Jan Villemson, "Time-Stamp -ing With Binary Linking Schemes", Crypto'98, http://home.cyber.ee/helger /papers/bllv98.html.

[3] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, "Handbook of Applied Cryptography", (July 1999), CRC Press, http://cacr. math.uwaterloo.ca/hac/.

[4] A. Michael Froomkin, "The essential role of Trusted Third Parties in Electronic Commerce", 75 Oregon L. Rev. 49, (1996),http://www.law.miami. edu/~froomkin/articles/trusted1.htm.

[5] Bruce Schneier, "Applied Cryptography - Protocols, Algorithms and Source Code in C", (1994), John Wiley & Sons.

[6] Chiara Bulgarelli, "Tecniche di Time-Stamping", (in italian), available at http://www.cs.unibo.it/~bulgarel/homepg/critto.html.

[7] Helger Lipmaa, "Digital Signatures and Authentication", version 2.0, (June 28 1999), Cybernetica, available at http://www.cyber.ee/infosecurity/resources/auth/.

[8] Henri Massias and Jean-Jacques Quisquater, "Time and Cryptography", Technical Report, Universite Catholique de Louvain, (March 1997), TIMESEC Technical Report WP1, http://www.dice.ucl.ac.be/crypto/TIMESEC/TIMESEC.html.

[9] Josh Benaloh and Michael de Mare, "Efficient Broadcast Time-Stamping", Microsoft Research Lab., http://www.research.microsoft.com/crypto/papers/tss.ps.

[10] Jeff Rothenberg, "Ensuring the Longevity of Digital Documents", Scientific American, January 1995.

[11] Mike Just, "Techniques for Digital Timestamping", available at http://www.scs.carleton.ca/~just/papers/CAMS97_slides.ps.

[12] NetscapeCommunications Corporation, "Introduction to PublicKey Cryptpgraphy", available at http://www.iplanet.com/developer/docs/articles/security/pki.html.

[13] Stuart Haber, Kaliski, W. Scott Stornetta, "How Do Digital Time -Stamps Support Digital Signatures?", Cryptobytes (The RSA security magazine), available at ftp://ftp.rsa.com/pub/cryptobytes/crypto1n3.pdf.

[14] Stuart Haber and Wakefield Scott Stornetta, "How to Time-Stamp a Digital Document", Journal of Cryptology, Vol. 3, No. 2, pp. 99-111 (1991).

[15] Stuart Haber and W. Scott Stornetta, "Secure names for bit-strings" (1997), available at http://www.star-lab.com/haber/dig-time-stamping.html

[16] TecLab presentation : "PKI: gli strumenti tecnologici" (in italian), available at http://www.seclab.com/publications/seclab-pki-4-99/sld001.htm.

저자 소개

정경원 (e-mail: kwjung09@hitel.net)
1983년: 숙명여자대학교 경영학과 졸업(학사)
1993년: 숙명여자대학교교육대학원 졸업(석사)
2000년: 숭실대학교 컴퓨터학과 수료(박사)
1995년~1998년 : 협성대학교 강사
1983년~1989 : 경상여자고등학교 교사
1989년~현재 : 세민정보산업고등학교 교사
관심분야 : 공개키 인증 구조, 데이터 통신, 네트워크 보안, 정보보안

도경화 (e-mail : khdo0905@dreamwiz.com)
1997년: 건양대학교 컴퓨터공학과 졸업(학사)
1999년: 숭실대학교 컴퓨터학과 졸업 (석사)
2002년: 숭실대학교 컴퓨터학과 수료(박사)
2001년~현재:숭실대학교 생산기술연구소 연구원
관심분야 : 공개키 인증 구조, 네트워크 보안(방화벽, IDS, VPN), 데이터 통신, 암호학

전문석 (e-mail : mjun@computing.ssu.ac.kr)
1980년: 숭실대학교 전자계산학과 졸업(학사)
1996년: Univ. of Maryland 전산과 졸업(석)
1989년: Univ. of Maryland 전산과 졸업(박사)
1989년:Morgan State Univ. 전산수학과 교수
1989년~1991년: NMSU. 부설PS랩 책임연구원
1991년~현재: 숭실대학교 정보과학대학 정교수
관심분야 : PKI, 정보보안, 보안프로토콜설계, 침입-차단 시스템, 암호학, 컴퓨터네트워크, 병렬처리.