Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.26, NO.6, Dec. 2016 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2016.26.6.1413

해시 기반 서명 기법 최신 기술 동향 및 전망*

박 태 환,† 배 봉 진, 김 호 원‡
부산대학교 전기전자컴퓨터공학과

Hash-Based Signature Scheme Technical Trend and Prospect*

Tae-hwan Park,† Bong-jin Bae, Ho-won Kim‡
Pusan National University Department of Electrical and Computer Engineering

요 약

최근의 양자컴퓨터 기술 발전과 PQCrypto2016에서 미국 NIST의 포스트 양자 암호 표준 공모사업 발표로 인해 포스트 양자 암호에 대한 관심과 연구가 활발히 이루어지고 있다. 양자 컴퓨터와 관련된 대표적인 알고리즘인 Grover 알고리즘과 Shor 알고리즘으로 인해, 현재 사용되고 있는 다양한 대칭키 암호와 이산대수 기반의 공개키 암호의 안전성 재고가 필요한 상황에서 양자 컴퓨터에도 강인한 암호인 포스트 양자 암호 연구의 필요성이 발생하였다. 본 논문에서는 다양한 포스트 양자 암호 중 해시 기반 서명 기법의 최신 기술 동향과 전망에 대해 알아본다.

ABSTRACT

In these days, there are a lot of research results on the Post-Quantum Cryptography according to developing of quantum computing technologies and the announcement of the NIST's Post-Quantum Cryptography standard project. The key size of the existing symmetric key block ciphers are needed to increase and the security of discrete logarithm based public key cryptography can be broken by Grover's algorithm and Shor's algorithm. By this reason, a lot of cryptologist and mathematician research on safe cryptography against the quantum computer which is called as the Post-Quantum Cryptography. In this paper, we survey on recent technical trend on the Hash-Based Signature Scheme which is one of the Post-Quantum Cryptography and suggest the prospect of the Hash-Based Signature Scheme.

Keywords: Post-Quantum Cryptography, Hash-Based Signature Scheme, WOTS, MSS

I. 서 론

PQCrypto2016에서 미국 NIST(National Institute of Standards and Technology)의 포스트 양자 암호 공모사업에 대한 발표가 있었으며, 양자 컴퓨터 기술의 발전과 양자 계산과 관련하여 임의의 함수 f(x)가 1인 경우에 대해 양자 계산으로 $O(N^{1/2})$ 회의 질의로 충분히 찾을 수 있다는 Grover 알고리즘과 합성수 N에 대해, $O((\log N)^3)$의 계산 시간 내에 N을 인수 분해 할 수 있는 Shor의 알고리즘으로 인해, 기존의 대칭키 암호 키 크기 증가가 필요하며, 이산 로그 및 이산 대수기반의 공개키 암호의 안전도에 공격이 가능한 상황이다. 이에 대해 전 세계의 많은 암호학자 및 컴퓨터 공학자들이 양자 컴퓨터 기술에 안전한 암호 기술을 개발하고자 2006년부터 PQCrypto 학회를 통한 포스트 양자 암호에 대한 연구가 시작되었으며, EU의 경우, SAFEcrypto 프로젝트가, 일본의 경우, CREST CryptoMath 프로젝트가 진행되고 있다.

Received(09. 23. 2016), Modified(11. 14. 2016), Accepted(11. 24. 2016)
* 본 연구는 2016년도 산업통상자원부의 재원으로 한국에너지기술평가원(KETEP)의 지원을 받아 수행한 연구 과제입니다. (No. 20152000000170)
* 본 논문은 2016년도 하계 학술대회에 발표한 우수논문을 개선 및 확장한 것임
† 주저자, pth5804@pusan.ac.kr
‡ 교신저자, howonkim@pusan.ac.kr(Corresponding author)

포스트 양자 암호 연구와 관련된 학회로는 ETSI (European Telecommunications Standards Institute)의 "Quantum Safe Cryptography", 미국 NIST의 "Cybersecurity in a Post Quantum World"와 PQCrypt 학회를 개최하여 연구를 진행하고 있다. 이러한 포스트 양자 암호 중 하나인 해시(Hash)기반 서명 기법은 해시 함수의 역상(pre-image), 제 2역상(2nd pre-image), 충돌 쌍(collision)에 대한 안전성에 기인하여 보안성 및 안전성을 제공한다. 해시 기반 서명 기법은 완전이진트리(Complete Binary Tree) 중 하나인 머클 트리(Merkle tree)를 사용하여 기존의 RSA, ECDSA와 같은 서명 기법이 가지는 단일의 비밀키/공개키 쌍이 단일의 문서에 대한 서명 및 검증이 가능한 단점을 극복하며, 단일의 비밀키/공개키 쌍으로 다중의 문서 (2^H개의 문서, H: 머클 트리(Merkle tree) 의 높이) 에 대한 서명, 검증이 가능하다는 장점을 가지고 있다. 본 논문에서는 포스트 양자 암호 중 하나인 해시 기반 서명 기법에 대한 최신 기술 동향 분석 및 전망에 대해 알아본다. 논문의 구성은 2장에서 해시 기반 서명 기법과 관련된 연구 내용 및 최신 기술 동향에 대해 살펴보고, 3장에서는 해시 기반 서명 기법 연구 전망에 살펴보며, 마지막 4장에서 본 논문의 결론을 맺는 순서로 구성된다.

II. 본 론

본 장에서는 해시 기반 서명 기법과 관련된 내용 및 최신 연구 동향에 대해 살펴본다.

2.1 해시 기반 서명 기법

해시 기반 서명 기법의 보안 및 안전성은 다음과 같은 해시 함수의 특성인 일방향성(One-wayness), 충돌 저항성(Collision resistance), 역상 안전성(pre-image resistance), 제2역상 안전성(2nd pre-image resistance)에 안전성을 기반으로 하고 있다.

기존의 RSA, ECDSA 기반의 서명 기법의 경우, 하나의 비밀키/공개키 쌍으로 단일의 문서에 대해서만 서명/검증이 가능한 단점을 가지고 있다. 해시 기반 서명 기법의 경우, 1979년 Ralph Merkle이 완전이진트리(Complete Binary Tree)의 일종인 머클 트리(Merkle tree)를 사용하여 하나의 비밀키/공개키 쌍으로 다중의 문서(2 H 개의 문서, H: 머클 트리(Merkle tree)의 높이) 를 서명/검증할 수 있는 구조를 제안 하였다 [1]. 이후 머클 트리(Merkle tree) 구조의 변형을 통한 보다 많은 문서에 대한 서명/검증이 가능한 구조에 대한 연구가 진행되었다.

2.2 Lamport-Diffie one-time 서명

해시 기반 서명 기법 에서 각각의 문서에 대한 서명/검증을 위해, one-time 서명 기법을 사용한다. 본 절에서는 해시 기반 서명 기법 에서 사용하는 one-time 서명 기법 중 Lamport-Diffie one-time 서명에 대해 알아본다. Lamport-Diffie one-time 서명은 256bit 크기의 해시 함수를 사용하며, 아래의 수식과 같이 n×2n bit string으로 구성된 서명키 X와 검증키 Y를 사용하며, 이 때, 균일 랜덤(Uniformly random) 값을 사용한다[1].

$$X=(x_{n-1}[0],x_{n-1}[1],...,x_0[0],x_0[1]) \in\{0,1\}^{(n,2n)} \quad (1)$$

$$Y=(y_{n-1}[0],y_{n-1}[1],...,y_0[0],y_0[1]) \in\{0,1\}^{(n,2n)}, y_i[j]=f(x_i[j]), 0\le i\le n-1, j=0,1 \quad (2)$$

이러한 서명키 X, 검증키 Y를 생성하는 과정에서 one-way function f의 2n회 호출이 요구된다[1].

Lamport-Diffie one-time 서명 생성 과정은 문서 M에 대해, 메시지 다이제스트 $g(M)=(d_{n-1},...,d_0)$와 검증키 X를 사용하여 아래의 서명 값을 생성한다.

$$\sigma=(x_{n-1}[d_{n-1}],...,x_0[d_0])\in\{0,1\}^{(n,n)} \quad (3)$$

Lamport-Diffie one-time 서명 검증 과정의 경우, 검증키 Y, 단 방향 함수 f를 사용하여 아래의 조건을 만족 하는지를 확인한다.

$$(f(\sigma_{n-1}),...,f(\sigma_0))=(y_{n-1}[d_{n-1}],...,y_0[d_0]) \quad (4)$$

이러한 Lamport-Diffie one-time 서명의 경우, 매우 효율적이라는 장점을 가지지만, 서명의 크기가 매우 크다는 단점을 가지고 있다.

2.3 Winternitz one-time 서명

앞서 설명한 Lamport-Diffie one-time 서명의 경우, 서명의 크기가 매우 크다는 단점을 가지고 있다. 이러한 문제점을 해결하기 위해 메시지 다이제스트의 일부 비트를 동시에 서명하는 Winternitz one-time 서명이 제안되었다[1]. Winternitz one-time 서명에 있어서 동시에 서명하고자 하는 비트 수를 나타내는 Winternitz parameter(w)를 2 이상의 수로 선택한 후, 선택된 Winternitz parameter(w)를 기반으로 아래의 수식을 통해, t_1, t_2, t를 계산한다.

$$t_1 = \lceil n/w \rceil \tag{5}$$

$$t_2 = \lceil (\lfloor \log_2 t_1 \rfloor + 1 + w)/w \rceil \tag{6}$$

$$t = t_1 + t_2 \tag{7}$$

Winternitz one-time 서명의 서명키 X와 검증키 Y는 아래의 식을 통해 생성된다.

$$X = (x_{n-1}, \dots, x_1, x_0) \in \{0,1\}^{(n,t)} \tag{8}$$

$$Y = (y_{n-1}, \dots y_1, y_0) \in \{0,1\}^{(n,t)}, \quad y_i = f^{2^w-1}(x_i), 0 \le i \le t-1 \tag{9}$$

서명키 X의 x_i는 균일 랜덤(uniformly random)한 값으로 설정하며, 키 생성 과정은 총 $t(2^w-1)$회의 일 방향 함수 f를 수행하며, t×n-bit 길이의 서명키/검증키를 생성한다.

Winternitz one-time 서명 생성 과정은 문서 M에 대한 메시지 다이제스트, $g(M) = (d_{n-1}, \dots, d_0)$의 길이가 Winternitz parameter(w)에 의해 나누어 질 수 있도록 0을 최소한의 개수로 d 값 앞에 패딩 과정을 수행한다. 패딩 과정을 거친 메시지 다이제스트 d는 t_1개의 string($d = b_{t-1} \| \cdots \| b_{t-t_1}$, $b_i \in \{0,1,\dots,2^w-1\}$)으로 나눈다. 다이제스트 d에 대한 checksum(c)을 아래의 식과 같이 계산한다.

$$c = \sum_{i=t-t_i}^{t-1} (2^w - b_i) \tag{10}$$

계산된 checksum(c)은 Winternitz parameter(w)에 의해 나누어질 수 있도록 0을 최소한의 개수로 c 값 앞에 패딩 과정을 수행한다. 패딩 과정을 거친 checksum(c)은 t_2개의 string ($c = b_{t_2-1} \| \cdots \| b_0$)으로 나눈다. 문서 M에 대한 메시지 다이제스트를 사용하여 계산된 d, checksum(c)과 서명키 X를 사용하여 아래의 수식에 따라 서명을 생성한다.

$$\sigma = (f^{b_{t-1}}(x_{t-1}), \dots, f^{b_1}(x_1), f^{b_0}(x_0)) \tag{11}$$

서명 생성 과정은 $t(2^w-1)$회의 일 방향 함수 f를 수행하며, 생성된 서명은 t*n-bit의 길이를 가진다. Winternitz one-time 서명 검증 과정은 서명 값 ($\sigma = (\sigma_{n-1}, \dots, \sigma_0)$)과 bit string ($b = (b_{t-1}, \dots, b_0)$)를 사용하여 아래의 수식의 조건을 만족하는지 확인하여 검증 과정을 수행한다.

$$(f^{2^w-1-b_{t-1}}(\sigma_{n-1}), \dots, f^{2^w-1-b_0}(\sigma_0)) = (y_{n-1}, \dots, y_0) \tag{12}$$

서명 값에 대한 검증이 맞다면, $\sigma_i = f^{b_i}(x_i)$조건을 만족하기 때문에 아래의 수식과 같이 서명 값의 검증 확인 가능하다.

$$f^{2^w-1-b_i}(\sigma_i) = f^{2^w-1}(x_i) = y_i \tag{13}$$

Winternitz one-time 서명 검증 과정에 있어서 최악의 경우, t개의 모든 항목에 대해서 수행이

Table 1. Comparison results between Lamport-Diffie one-time signature and Winternitz one-time signature(b: security level, w: winternitz parameter, m: length of message) [6]

	Lamport-Diffie Signature	Winternitz one-time signature
signing key size	2bm	~2bm/log w
verification key size	2bm	~2bm/log w
signature size	bm	~2bm/log w
key-pair generation time	~2m	~wm/log w

필요하므로, $t(2^w-1)$회의 일 방향 함수 f를 수행해야 한다[1].

앞서 살펴본 Lamport-Diffie one-time 서명과 Winternitz one-time 서명의 서명/검증키, 서명 크기 및 키 생성 시간은 아래의 표와 같이 나타낼 수 있다.

Winternitz one-time 서명의 function chain의 경우, 해시함수를 사용하여 아래의 수식과 같이 구성될 수 있다.

$$c^i(x)=h_k(c^{i-1}(x))$$
$$=\underbrace{h_k\circ h_k\circ\cdots\circ h_k(x)}_{i-TIMEs} \quad (14)$$
$$,x\in\{0,1\}^n,c^0(x)=x$$

이러한 Winternitz one-time 서명(WOTS)의 변형으로는 $WOTS^{\$}$, $WOTS^{+}$가 있으며, 기존 WOTS의 안전성은 충돌 저항성 (Collision resistance)계열의 undetectable 일방향성 함수에 기반하며, $WOTS^{\$}$, $WOTS^{+}$의 경우, 각각 의사난수 함수(pseudorandom function)와 제 2역상 안전성(2nd pre-image resistance)에 안전성을 기반하고 있다. 이러한 $WOTS^{\$}$, $WOTS^{+}$의 function chain은 아래의 수식으로 표현될 수 있다.

$$WOTS^{\$}: c^i(x)=h_{c^i(x)}(r) \quad (15)$$

$$WOTS^{+}: c^i(x)=h_k(c^{i-1}(x)\oplus r_i) \quad (16)$$

2.4 Merkle tree Signature Scheme(MSS)

머클 트리 서명 기법(Merkle tree Signature Scheme, MSS)은 완전이진트리(Complete Binary tree)중 하나인 머클 트리(Merkle tree) 구조를 사용하여 트리의 높이인 H(H∈N, H≥2)에 따라 2^H개의 문서에 대한 서명/검증이 가능하다. 머클 트리 서명 기법(Merkle tree Signature Scheme, MSS)에서의 공개키는 머클 트리(Merkle tree)의 root에 위치하며, 2^H개의 개인키(서명키)는 leaf 노드에 위치한다. i번째 Leaf 노드에는 각각의 문서에 대한 개인키(서명키) 및 문서의 다이제스트 값($g(M_i),0\le i\le 2^H$)을 가지고 있다. 머클 트리(Merkle tree)의 inner 노드는 자신의 왼쪽, 오른쪽 자식 노드의 Concatenation 결과에 대한 해시 값($v_h[j]=g(v_{h-1}[2j]|v_{h-1}[2j+1]),1\le h\le H,0\le j\le 2^{H-h}$)을 가진다[1].

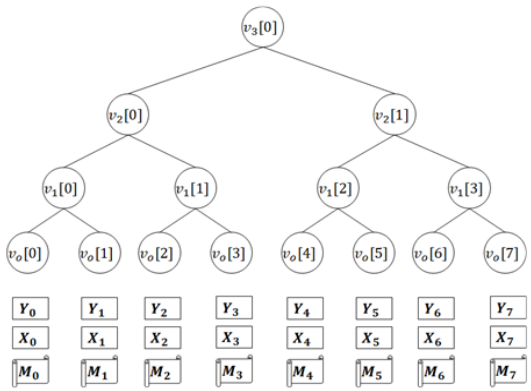

Fig. 1. Merkle tree Signature Scheme(merkle tree height, H=3)

Merkle tree의 서명키/검증키 생성 과정은 총 2^H개의 Winternitz one-time 서명키/검증키 쌍을 생성하며, 2^{H+1}-1회의 해시 함수 수행이 필요하다. 특히 머클 트리(Merkle tree)의 root에 위치한 공개키에 대한 효율적인 생성을 위해서 아래의 트리해시(Treehash) 알고리즘을 사용하여 최대 H개의 트리 노드만 저장함으로써 메모리 사용의 효율성을 높일 수 있다[1].

알고리즘 1.에서 보는 바와 같이, 노드 저장을 위해, STACK의 pop, push 연산을 사용하며, LEAFCALC(j) 연산은 j번째 leaf 노드에 대한 연산으로 j번째 one-time 검증키로부터 j번째 노드를 계산하는 과정을 의미한다.

머클 트리 서명 기법(Merkle tree Signature Scheme, MSS)의 서명 생성 과정은 앞서 생성한 one-time 서명키를 연속적으로 사용하며, 문서 M

Algorithm 1. Treehash Algorithm[1]

INPUT: Merkle tree Height h
OUTPUT: root of Merkle tree

for $j=0,\dots,2^H-1$ **do**
NODE1←LEAFCALC(j)
While NODE1.h==STACK.pop().h **do**
I. NODE2←STACK.pop()
II.NODE1←g(NODE2||NODE1)
III. STACK.push(NODE1)
R ← STACK.pop(NODE1)
return R

에 대한 서명을 위해, 문서 M에 대한 해시 값(다이제스트)인 g(M) 먼저 계산한 이후, 서명키 $X_s(s \in 0, \ldots, 2^H-1)$를 사용하여 one-time 서명인 σ_{OTS}를 생성한다. 머클 서명(Merkle Signature)에는 one-time 서명과 이에 대응하는 검증키 Y_s를 포함한다. 그리고 서명에 대한 검증을 위해, 검증자에게 제공할 인덱스 s와 Y_s의 인증 패스(Authentication Path)인 $A_s = \{a_0, \ldots, a_{H-1}\}$을 제공한다. 인증 패스(Authentication Path) 상의 노드 h는 인증 경로 상에서 높이가 h인 노드의 형제 노드를 의미하며, 계산 공식은 아래와 같다[1].

$$a_h = v_h\left[\lceil (s/2^h)-1 \rceil\right], \quad \text{if } \lfloor s/2^h \rfloor \equiv 1 \bmod 2, h = 0, \ldots, H-1 \tag{17}$$

$$a_h = v_h\left[\lceil (s/2^h)+1 \rceil\right], \quad \text{if } \lfloor s/2^h \rfloor \equiv 0 \bmod 2, h = 0, \ldots, H-1 \tag{18}$$

아래의 그림은 H=3인 Merkle tree를 사용한 Merkle tree Signature Scheme에서 인덱스 s가 3인 경우, 서명 생성 과정을 나타낸다. s가 3인 경우, 아래의 수식과 같은 머클 서명(Merkle Signature) 을 가진다.

$$\sigma_s = \{s, \sigma_{OTS}, Y_s, (a_0, \ldots, a_{H-1})\} = \{3, \sigma_{OTS}, Y_3, (a_0, a_1, a_2)\} \tag{19}$$

머클 트리 서명 기법(Merkle tree Signature Scheme, MSS)의 서명 검증 과정은 아래와 같이 크게 2단계로 나누어진다[1].

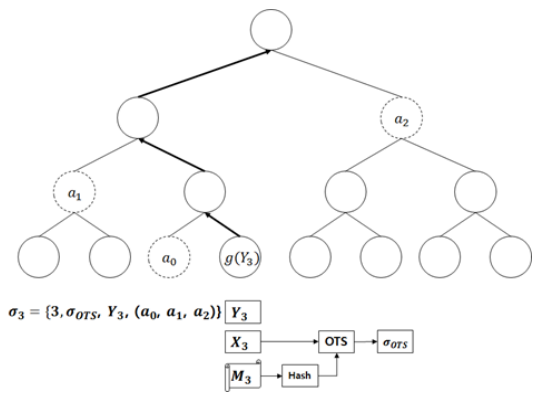

Fig. 2. Merkle tree Signature generation(index s=3, merkle tree height H=3)

1단계: 검증자가 one-time 검증키 Y_s를 사용하여 서명 σ_{OTS}에 대한 검증 과정

2단계: 검증자가 s번째 leaf 노드인 $g(Y_s)$로부터 머클 트리(Merkle tree)의 root까지의 계산 경로 $(p_0, \ldots, p_{H-1})$와 인증 경로$(a_0, \ldots, a_{H-1})$를 이용하여 one-time 검증키 Y_s에 대한 검증 과정이며, 검증 과정 식은 아래와 같다.

$$p_h = g(a_{h-1} \| p_{h-1}), p_0 = g(Y_s), \quad \text{if } \lfloor s/2^h \rfloor \equiv 1 \bmod 2, h = 0, \ldots, H \tag{20}$$

$$p_h = g(p_{h-1} \| a_{h-1}), p_0 = g(Y_s), \quad \text{if } \lfloor s/2^h \rfloor \equiv 0 \bmod 2, h = 0, \ldots, H \tag{21}$$

인덱스 s는 인증 패스 노드들의 순서를 정하는데 사용되며, 위의 검증 과정을 통해 얻게 되는 p_h가 머클 트리(Merkle tree)의 root에 위치한 공개키와 동일 한 경우, one-time 검증키 Y_s에 대한 검증이 완료된다.

2.5 해시 기반 서명 알고리즘 관련 연구 동향

본 절에서는 해시 기반 서명 기법과 관련한 최신 연구 동향에 대해서 살펴본다. Rohde, Sebastian, et al.[2]에서는 8비트 스마트카드에 적합한 MSS(Merkle signature scheme)를 제시하며, RSA, ECDSA에 비해 적은 코드 사이즈로 대등한 성능을 보인다.

Buchmann, Johannes, et al.[3]에서는 MSS 알고리즘에 비해 개인키 사이즈, 키 쌍 생성 시간, 서명 생성 시간을 줄인 CMSS를 개발하였다. 본 논문에서는 Java Cryptographic Service Provider인 FlexiProvider를 이용해 구현하였으며, Microsoft Outlook에서 메시지 서명을 예로 제시하였다.

De Oliveira method.[4]에서는 HBS의 알고리즘 중 MSS, GMSS, XMSS, XMSS-MT 알고리즘에서 multi-buffer 기법을 이용하여 키 생성, 서명, 검증 과정을 고속화하였으며, XMSS의 경우 128비트 보안 강도에서 SHA2-256을 사용했을 때, 2,001,479 cycles의 성능을 보였다[4].

Eisenbarth, Thomas, et al.[5]에서는 인증 경로 연산에서의 state-of-the-art 알고리즘을 통한 연산 고속화와 AVR-ATxmega 계열 보드에서

해당 알고리즘을 구현하여 이론뿐만 아니라, 실험적으로도 하드웨어 상에서 가속화된 성능을 보여줬다.

Hülsing, Andreas, et al.[6]에서는 XMSS와 Multi-tree XMSS 등과 같은 알고리즘에 대한 확장 방향성 및 stateless 방식에 대한 outline을 제시하고 있다.

Bernstein, Daniel J., et al.[7]에서는 stateless방식의 해시 기반 서명 기법인 SPHINCS를 제안하였다. 본 논문에서 제안한 SPHINCS는 hyper-tree구조를 적용하여 여러 개의 머클 트리(Merkle tree)를 연결한 구조를 사용하며, 인덱스의 경우, 의사 난수 값으로 선택하여 처리함으로써 기존의 상태(state)방식의 해시 기반 서명 기법에 비해 빠른 성능을 보인다. SPHINCS에서는 내부 트리 구조에서 앞서 설명한 $WOTS^{+}$를 one-time 서명으로 사용하며, 보안강도 및 효율성을 높이기 위해, Few-Time Signature 기법 중 하나인 HORST를 사용한다. 논문에서 제시한 SPHINCS-256은 양자컴퓨터에 대한 128비트 보안 강도를 제공하며, 높이가 5인 머클 트리(Merkle tree) 12개로 구성된다. SPHINCS-256에서의 Winternitz parameter (w)는 16으로 설정한 $WOTS^{+}$를 적용하였다. 이를 통해, SPHINCS-256의 서명 크기는 41KB, 공개키 1KB, 비밀키 1KB의 크기를 제공하며, Intel Xeon E3-1275프로세서(3.5GHz) 상에서 AVX2 기반 SIMD 구현 결과를 제시하고 있다.

Hülsing, Andreas, et al.[8]에서는 ARM Cortex M3환경 상에서의 stateless방식의 해시 기반 서명 기법인 SPHINCS-256과 XMSS-MT의 구현 결과(SPHINCS-256의 경우, 729,942,616cycles, XMSS-MT의 경우, 22,725,616 cycles)를 제시하고 있다.

III. 전 망

해시기반 서명 알고리즘의 경우, 아직 최적화 연구와 키/서명 크기 최소화 그리고 상태(state)를 사용하지 않는 SPHINCS[7]에 대한 최적화 및 적용 연구가 필요한 상황이다.

XMSS(eXtended Merkle tree Signature Scheme)의 경우, 제 2역상 공격에 대한 강인성을 가지며, 현재 IETF 표준의 파이널 라운드에 올라가 있는 상황이며, 이에 대한 연구가 활발히 이루어질 것으로 전망된다. 현재의 해시기반 서명 알고리즘의 키/서명 크기가 크다는 단점이 있기 때문에 키/서명 크기 최소화 연구와 서명/검증과정에 대한 속도 최적화 연구가 활발히 이루어질 것이다. 상태(state)를 사용하는 기존의 해시 기반 서명 기법에서의 낮은 연산 속도를 개선하기 위해 제안된 stateless 해시 기반 서명 기법인 SPHINCS[7]에 대한 최적화 연구 및 다양한 환경으로의 적용 연구가 앞으로 필요하며, 활발히 이루어질 것으로 예측된다. 이러한 연구와 더불어 각국에서 보유한 해시함수 관련 표준과의 연동 및 적용에 대한 연구를 통해, NIST의 포스트 퀀텀 암호 공모전에 대한 준비 또한 중요할 것으로 예상된다.

IV. 결 론

본 논문에서는 해시 기반 서명기법에 대한 최신 기술 동향 분석 및 전망에 대해 알아보았다. 현재의 가장 큰 문제는 사용되는 키 크기와 서명 크기가 매우 크다는 단점과 서명과 검증 과정에서의 시간이 오래 걸리는 단점이 있으며, 이에 대한 키/서명 크기 최소화 및 알고리즘 속도 최적화에 대한 연구가 필요할 것이며, state없이 동작하는 stateless Hash 기반 서명인 SPHINCS에 대한 연구와 사용되는 해시함수에 대한 최적화 또한 필요할 것으로 보인다. 본 논문의 동향 및 전망은 향후 해시 기반 서명 기법 연구에 활용될 수 있을 것이라고 생각된다.

References

[1] Bernstein, Daniel J., Johannes Buchmann, and Erik Dahmen, eds, "Post-quantum cryptography," Springer Science & Business Media, pp. 35-91, Nov. 2008.

[2] Sebastian Rohde, "Fast Hash-Based Signatures on Constrained Devices," CARDIS 2008, pp. 104-117, Sep. 2008.

[3] Johannes Buchmann, "CMSS - An Improved Merkle Signature Scheme," Progress in Cryptology - INDOCRYPT, pp. 349-363, Dec. 2006.

[4] Ana Karina D.S. de Oliveira, "An Efficient Software Implementation of the

Hash-Based Signature Scheme MSS and Its Variants," LATINCRYPT 2015, pp. 366-383, Aug. 2015.
〔5〕 Thomas Eisenbarth, "A Performance Boost for Hash-based Signatures," LNCS 8260, pp. 166-182, Jan. 2013.
〔6〕 Andreas Hülsing, "Hash-based Signatures: An Outline for a New Standard," NIST Workshop on Cybersecurity in a Post-Quantum World, pp. 1-12, Apr. 2015.
〔7〕 Bernstein, Daniel J., et al, "SPHINCS: practical stateless hash-based signatures," Advances in Cryptology-EUROCRYPT 2015, Springer Berlin Heidelberg, pp. 368-397, Apr. 2015.
〔8〕 Hülsing, Andreas, Joost Rijneveld, and Peter Schwabe, "ARMed SPHINCS Computing a 41KB signature in 16KB of RAM," Public-Key Cryptography - PKC 2016, pp. 446-470, Mar. 2016.

〈저자소개〉

박 태 환 (Tae-hwan Park) 학생회원
2013년 2월: 부산대학교 정보컴퓨터공학부 학사 졸업
2013년 3월~현재: 부산대학교 전기전자컴퓨터공학과 석, 박사 통합과정
〈관심분야〉 암호화 구현, IoT 디바이스 보안, Post-Quantum Cryptography

배 봉 진 (Bong-jin Bae) 학생회원
2015년 2월: 부산대학교 정보컴퓨터공학부 학사 졸업
2015년 8월~현재: 부산대학교 전기전자컴퓨터공학과 석사과정
〈관심분야〉 암호화 구현, IoT 디바이스 보안, Post-Quantum Cryptography

김 호 원 (Ho-won Kim) 종신회원
1993년 2월: 경북대학교 전자공학과 학사 졸업
1995년 2월: 포항공과대학교 전자전기공학과 석사 졸업
1999년 2월: 포항공과대학교 전자전기공학과 박사 졸업
2008년 2월: 한국전자통신연구원 정보보호연구단 선임연구원/팀장
2008년 3월~현재: 부산대학교 전기컴퓨터공학부 부교수
〈관심분야〉 스마트그리드 보안, RFID/USN 정보보호 기술, PKC 암호, VLSI 설계, embedded system 보안, IoT