논문 22-47-07-01 The Journal of Korean Institute of Communications and Information Sciences '22-07 Vol.47 No.07
https://doi.org/10.7840/kics.2022.47.7.909

Wave 부호 기반 전자서명 기법의 효율적 검증 연구

임 성 창•, 최 성 봉*, 조 기 환**, 이 형 태°

A Study on Efficient Verification of Wave Code-Based Signature

Seongchang Im•, Seongbong Choi*, Gihwan Cho**, Hyung Tae Lee°

요 약

자율주행 차량 네트워크와 같이 전송된 데이터의 무결성을 실시간으로 보장해야 하는 상황이 증가함에 따라 전자서명의 서명뿐만 아니라 검증에 소요되는 시간이 중요해지고 있다. 본 논문에서는 부호기반 전자서명인 Wave의 효율적인 검증 방법에 대하여 연구한다. 이를 위하여, 먼저 Wave 전자서명의 검증 과정에 최근 Boschini, Fiore, Pagnin이 제시한 전자서명 검증 알고리즘의 효율적인 검증 방법으로의 변환 방법을 적용할 수 있음을 확인한다. 또한, 적용한 결과의 구현을 통하여 온라인 검증 시간의 성능 향상 효과를 살펴본다. 본 논문의 구현 결과에 따르면, 128비트 안전성을 갖는 Wave 전자서명에 대하여 오프라인 계산을 수행해두면, 이를 이용하여 온라인 검증 속도를 기존의 Wave 전자서명에 비해 약 12배 향상시킬 수 있음을 확인한다.

키워드 : 전자서명, 부호 이론, Wave 전자서명, 효율적 검증, 온라인 검증
Key Words : Digital signatures, coding theory, Wave signatures, efficient verification, online verification

ABSTRACT

In this paper, we study an efficient verification way of Wave code-based signatures. To this end, we first identify that the recent generic transformation for efficient verification, proposed by Boschini, Fiore, and Pagnin, can be applied to the verification algorithm of Wave signatures. Then, we provide various implementation results of efficient verification of Wave signatures. According to our experimental results, for 128-bit security, the online phase of our efficient verification outperforms that of the original Wave signatures by a factor of about 12 times, with the help of additional offline computation.

Ⅰ. 서 론

전자서명(Digital Signature)은 전송되는 데이터의 무결성(Data Integrity)을 보장해주는 암호 기법의 하나로, TLS(Transport Layer Security), 공인인증서, 블록체인, 자율주행 통신 등 보안 분야 전반에 걸쳐 다양하게 활용되고 있다. 1977년 Rivest, Shamir, Adleman (RSA)[1]이 인수분해 기반의 최초의 전자서명을 제시한 이후, 다양한 전자서명 기법이 제시되었으며, 최근에는 양자컴퓨터의 개발에 대비하여 격자[2], 다변수 다항식[3], 해시함수[4], 부호[5,6] 등 다양한 암호학적 도구를 활용하여 양자컴퓨팅에 안전한 전자서명이 제시되고 있다. 그러나 대부분의 전자서명 설계에 관한 연구는 안전하고 효율적인 서명 생성 알고리즘

※ 본 연구는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행되었습니다. (No. NRF-2021R1A2C1007484)
• First Author : Department of Computer Science and Engineering, Jeonbuk National University, pabi3564@naver.com, 정회원
° Corresponding Author : School of Computer Science and Engineering, Chung-Ang University, hyungtaelee@cau.ac.kr, 정회원
* School of Computer Science and Engineering, Chung-Ang University, welq2st@cau.ac.kr
** Department of Computer Science and Engineering, Jeonbuk National University, ghcho@jbnu.ac.kr, 정회원
논문번호 : KICS202203-046-A-RU, Received March 28, 2022; Revised May 15, 2022; Accepted May 20, 2022

의 설계에 초점을 맞추고 있으며, 효율적인 검증 알고리즘에 관한 연구는 제한적이다.

계산 능력이 제한된 기기의 사용, 실시간 검증이 필요한 환경 등 전자서명의 응용 분야가 다양해짐에 따라 전자서명의 서명 생성 알고리즘뿐만 아니라 검증 알고리즘의 효율성도 중요시되고 있다. 한 예로 자율주행 차량 통신 네트워크상에서 도로를 주행 중인 두 차량이 주행 정보를 송·수신하는 과정을 가정하자. 이때, 두 차량 사이에 주고받는 주행 관련 데이터에 위·변조가 일어나지 않았음을 보장하는 과정은 필수적이며, 해당 데이터는 실시간으로 주행에 반영되어야 한다. 따라서 전송받은 서명의 검증은 최대한 빠르게 수행되어야 한다.

효율적인 검증 알고리즘을 위해 전자서명 자체를 새롭게 설계하는 대신 기존의 전자서명을 변형하는 새로운 접근 방법이 최근 주목을 받고 있다. 2019년 Le, Kelkar, Kate[7]는 전자서명의 검증 과정 전체를 진행하는 대신 일부분만 수행하여 확률적으로 검증을 수행하는 탄력적인 전자서명(Flexible Signature) 기법의 개념을 제시하고 해시함수 기반의 탄력적인 전자서명을 제안하였는데, 기존 검증 알고리즘의 경우 검증의 모든 과정을 수행해서 1 또는 0의 결론을 내린 반면, 제안된 기법은 확률적으로 검증 결과를 나타낸다. 2021년 Taleb과 Vergnaud[8]는 Le 등의 아이디어를 확장하여 현재 널리 사용되고 있는 전자서명인 RSA 전자서명[1]과 타원곡선 전자서명(ECDSA, Elliptic Curve Digital Signature Algorithm)[9], 그리고 대표적인 격자 기반 전자서명인 Gentry, Peikert, Vaikuntanathan (GPV)의 전자서명[2], 부호 기반 전자서명 Wave[5]에 적용하였다. 나아가, 최근 Boschini, Fiore, Pagnin (BFP)[10]은 검증 알고리즘이 특정 방정식의 형태를 만족시키는 전자서명이 주어지면, 키 생성 단계의 사전계산을 이용하여 온라인 검증 시간을 줄일 수 있는 효율적인 검증 방법을 갖도록 하는 일반적인 변환 방법을 제시하였으며, 이를 이용하여 격자 기반의 GPV 전자서명, 다변수 다항식 기반의 Rainbow 전자서명[3]에 적용하였다. 한편, 부호 기반 전자서명의 경우에는 후속 연구로 남겨두었다.

본 연구에서는 최근 제시된 부호 기반 전자서명인 Wave 전자서명에 대하여 BFP의 방법을 적용하여 효율적인 검증 알고리즘을 갖는 부호기반 전자서명 기법을 얻고자 한다. BFP 방법을 적용하기 위해서는 검증 알고리즘이 공개키를 이용하여 생성되는 행렬 M, 서명을 이용하여 생성되는 벡터 v, 메시지의 해시함수 값을 이용하여 생성되는 벡터 u에 대하여 $Mv = u$를 만족하여야 한다. 본 논문에서는 먼저 Wave 전자서명의 검증 과정을 위의 식으로 이해할 수 있음을 살펴본다. 나아가 BFP 방법을 적용하여 얻은 Wave 전자서명의 효율적인 검증 과정에 대한 실험 결과를 제시한다. 본 논문의 구현 결과에 따르면, 128비트 안전성을 달성하는 파라미터 하에서 기존의 Wave 전자서명의 검증이 약 1ms가 소요되지만, BFP 방법을 적용하여 얻은 효율적인 검증 방법의 경우 오프라인 단계에서 추가로 6,653ms의 시간을 이용하여 계산해두면, 약 0.091ms가 소요되어 12배가량 온라인 검증 시간을 단축할 수 있게 된다.

본 논문의 2장에서는 Wave 전자서명에 대하여 살펴보며, 3장에서는 전자서명의 효율적인 검증을 위한 기존 연구들에 관하여 소개하고, BFP변환 방법에 대하여 구체적으로 살펴본다. 4장에서는 Wave 전자서명에 BFP 방법의 적용 가능 여부에 관하여 확인하고, 실제 적용하여 구현한 여러 실험 결과를 제시한다.

Ⅱ. Wave 부호 기반 전자서명

이번 장에서는 Wave 전자서명 기법에 대하여 살펴보도록 한다. Wave 전자서명 기법은 Debris-Alazard, Sendrier, Tillich에 의해 2019년 제시된 기법[5]으로 해시 후 서명(Hash-and-Sign) 기법을 이용하여 설계되었다. 따라서 같은 해시 후 서명 기법을 활용하는 Courtois, Finiasz, Sendrier가 제시한 부호 기반 서명인 CFS 전자서명[11]과 유사한 구조를 갖지만, CFS 전자서명이 Goppa 부호를 이용하는 반면, Wave 전자서명은 정규화 및 일반화된 $(U, U+V)$ 부호를 활용하게 됨에 따라 CFS 전자서명에 비해 서명 알고리즘의 속도를 높이고, 파라미터의 크기를 줄이는 장점을 얻게 되었다.

일반적인 전자서명 기법과 같이 Wave 전자서명도 키 생성 알고리즘, 서명 알고리즘, 검증 알고리즘으로 구성되어 있으며, 구체적인 알고리즘은 다음과 같다.

2.1 Wave의 키 생성 알고리즘

Wave 부호 전자서명의 키 생성 알고리즘은 정규화 및 일반화된 $(U, U+V)$ 부호라는 특별한 형태를 만족하는 부호를 이용하여 설계되었다. 키 생성 알고리즘을 알아보기 위하여, $(U, U+V)$ 부호를 먼저 소개하고 이를 바탕으로 구체적인 키 생성 알고리즘을 살펴보도록 한다.

2.1.1 정의(UV-정규화)

q개의 원소로 이루어진 유한체(finite field) F_q와 짝 수 n에 대하여 a, b, c, d를 각각 차원이 $n/2$인 $F_q^{n/2}$ 상의 벡터라 하자. 이 때, 함수 $\psi_{a,b,c,d}$를 다음과 같이 정의한다.

$$\psi_{a,b,c,d} : F_q^{n/2} \times F_q^{n/2} \rightarrow F_q^{n/2} \times F_q^{n/2}$$
$$(x, y) \mapsto (a \odot x + b \odot y, c \odot x + d \odot y)$$

위의 식에서 두 벡터 $x = (x_1, x_2, \cdots, x_{n/2})$와 $y = (y_1, y_2, \cdots, y_{n/2})$에 대하여

$$x \odot y = (x_1 y_1, x_2 y_2, \cdots, x_{n/2} y_{n/2})$$

로 계산한다. 만약 모든 $1 \le i \le n/2$에 대하여 $a_i d_i - b_i c_i = 1$, $a_i c_i \neq 0$을 만족하면 우리는 함수 $\psi_{a,b,c,d}$가 UV-정규화되어 있다고 한다.

각각의 원소에서 정의된 함수의 UV-정규화는 두 개의 $F_q^{n/2}$의 부분 공간(subspace) U, V에 대해서도 확장하여 적용할 수 있으며, 다음과 같이 정의된다.

$\psi_{a,b,c,d}(U, V) = \{\psi_{a,b,c,d}(u, v) \mid u \in U, v \in V\}$

위의 성질을 이용하여 우리는 새로운 부호를 정의할 수 있다.

2.1.2 정의(정규화-일반화된 $(U, U+V)$-부호)

n을 짝수인 정수라 하고, $\psi = \psi_{a,b,c,d}$를 UV-정규화 함수라 하자. 패리티 검증 행렬(parity check matrix)을 각각 H_U, H_V로 갖는 $F_q^{n/2}$ 상의 두 개의 벡터 공간에 대하여 $\psi(U, V)$를 정규화-일반화된 $(U, U+V)$-부호라 한다. 정규화-일반화된 $(U, U+V)$-부호는 그 차원으로 $\dim U + \dim V$를 갖고 대응되는 패리티 검증 행렬은

$$H(\psi, H_U, H_V) = \begin{pmatrix} H_U D & -H_U B \\ -H_V C & H_V A \end{pmatrix} \quad (1)$$

가 되는데, A, B, C, D는 각각 a, b, c, d의 값을 대각선 값으로 갖고 나머지 값은 모두 0인 대각화 행렬(diagonal matrix)이다.

Wave 전자서명의 키 생성 알고리즘은 상술한 정규화-일반화된 $(U, U+V)$-부호를 안전성 파라미터의 크기에 맞춰 임의로 생성하여 이를 비밀키로 사용하며, 가역행렬(invertible matrix)과 치환행렬(permutation matrix)을 임의로 생성하여 이를 곱하여 공개키를 생성한다. 구체적인 알고리즘은 다음과 같다.

1) Wave.Keygen(λ): 안전성 파라미터 λ를 입력값으로 받아 다음의 과정을 수행한다.
   (1) 안전성 파라미터에 맞는 부호가 정의될 유한체 F_q, 부호의 길이 n, 차원 k, 해밍 웨이트(Hamming weight) 파라미터인 w를 각각 정하고, $k = k_U + k_V$를 만족하는 값 k_U, k_V를 정한다.
   (2) 패리티 검증행렬 H_U와 H_V를 임의로 선택한다. 이 때, H_U와 H_V는 각각 F_q상에서 정의된 $(\frac{n}{2} - k_U) \times \frac{n}{2}$ 행렬과 $(\frac{n}{2} - k_V) \times \frac{n}{2}$ 행렬이다.
   (3) 임의의 UV-정규화 함수 ψ를 선택한다.
   (4) 임의의 가역행렬 S를 선택한다.
   (5) 임의의 치환행렬 P를 선택한다.
   (6) 2단계와 3단계에서 선택한 ψ, H_U, H_V를 이용하여 정의 2.1.2에서 제시된 식 (1)을 이용하여 정규화-일반화된 $(U, U+V)$-부호의 패리티 검증 행렬 $H_{sk} = H(\psi, H_U, H_V)$를 계산한다.
   (7) $H_{pk} = SH_{sk}P$를 계산한다.
   (8) $pk = H_{pk}$와 $sk = (H_{sk}, S, P)$를 각각 공개키와 비밀키로 출력한다.

2.2 Wave의 서명 생성 알고리즘

이번 절에서는 Wave 전자서명의 서명 생성 알고리즘에 대하여 살펴본다. 2.1절에서 소개한 정규화-일반화된 $(U, U+V)$-부호의 사용은 효율적인 디코딩 알고리즘을 제공하여 Wave 전자서명의 서명 생성 알고리즘의 효율을 높여준다. 본 논문의 경우, 검증 알고리즘에 초점을 맞추고 있어 서명 생성에 사용되는 디코딩 알고리즘의 구체적인 내용은 생략하도록 하며, 패리티 검증 행렬 H_{sk}에 대하여 디코딩 알고리즘 $Dec_{H_{sk}}$가 주어졌다고 가정한다. 해당 디코딩 알고리즘의 자세한 내용은 Wave 전자서명의 논문[5]에서 확인할 수 있다.

1) Wave.Sign(sk, m): 비밀키 $sk=(H_{sk}, S, P)$와 메시지 m을 입력 값으로 받아 다음을 수행한다.
   (1) 임의의 값 r을 $\{0,1\}^{\lambda}$에서 선택한다.
   (2) 해시함수 $Hash$에 대하여 $s=Hash(m,r)$을 계산한다.
   (3) 비밀키 sk에 포함된 S를 이용하여 $s(S^{-1})^T$를 계산한 후, 이를 입력 값으로 하여 디코딩 알고리즘 $Dec_{H_{sk}}(s(S^{-1})^T)$를 수행하여 e를 얻는다. (본 논문에서 행렬 A에 대하여 A^T는 A의 전치행렬(transpose matrix)을 의미한다.)
   (4) 메시지 m에 대한 서명 $\sigma=(eP, r)$을 출력한다.

2.3 Wave의 검증 알고리즘

이번 절에서는 Wave 전자서명의 검증 알고리즘을 소개하고, Wave 전자서명의 정확성(correctness)에 대해서 살펴본다. Wave 전자서명의 검증 알고리즘은 다음과 같다.

1) Wave.Verify(pk, σ): 공개키 pk와 서명 $\sigma=(eP, r)$을 입력 값으로 받아 다음을 수행한다.
   (1) 검증하고자 하는 메시지 m과 해시함수 $Hash$에 대하여 $s'=Hash(m,r)$을 계산한다.
   (2) $e'=eP$의 웨이트가 w이고, $e'(H_{pk})^T=s$가 성립하면 1(Accept), 그렇지 않으면 0(Reject)을 출력한다.

위의 검증 과정에서 $e'=eP$가 성립한다면,

$$e'(H_{pk})^T = e'(SH_{sk}P)^T = e'P^T H_{sk}^T S^T \\ = ePP^T H_{sk}^T S^T = eH_{sk}^T S^T \quad (2)$$

를 만족한다. 이 때, 네 번째 등식은 P가 치환행렬이므로 $PP^T=I$ (I는 단위행렬(identity matrix))로부터 성립하게 된다. 또한, e는 서명 생성 알고리즘의 3단계로부터 $Dec_{H_{sk}}(s(S^{-1})^T)$의 출력 값이므로, $eH_{sk}^T = s(S^{-1})^T$가 성립한다. 이를 (2)에 대입하면, $eH_{sk}^T S^T = s(S^{-1})^T S^T = s$ 가 성립한다. 한편, P는 치환행렬로 벡터의 웨이트를 보존하므로 벡터 e'에 대하여 $wt(e')=wt(eP)=wt(e)=w$를 만족하게 되어 검증에 성공한다. (이 때, $wt(x)$는 벡터 x의 해밍 웨이트이다.)

Ⅲ. 전자서명의 효율적인 검증을 위한 기존 연구 및 BFP 변환 방법

2장에서 Wave 전자서명[5]의 키 생성, 서명 생성, 검증 알고리즘을 살펴보았다. 이번 장에서는 사전계산을 이용한 전자서명의 효율적인 검증에 관한 기존 연구를 소개하고, 이 중 최근 제시된 BFP 변환 방법[10]에 대하여 살펴본다.

3.1 전자서명의 효율적인 검증 관련 기존 연구

전자서명 기법에서 효율적인 서명 생성을 위한 연구는 활발히 이루어졌지만, 전자서명의 효율적인 검증에 관한 연구는 상대적으로 충분히 진행되지 않았다. 그러나 사람의 안전과 직결될 수 있는 차량 네트워크 상의 데이터 무결성 검증, 블록체인에서 신속한 데이터 처리를 위한 효율적인 서명 검증 등 검증 속도가 중요시되는 전자서명의 응용들이 다수 등장함에 따라 전자서명의 효율적인 검증에 관한 연구가 점차 주목받고 있다.

효율적인 전자서명의 검증을 위해서 먼저 확률적인 검증 방법이 제시되었다. 전형적인 전자서명 검증 알고리즘은 양자택일(all-or-nothing)의 관점에서 성공과 실패 중 하나를 택하는 방식이며, 이를 위해서는 전자서명의 검증 알고리즘 전체를 수행해야만 한다. Le, Kelhar, Kate[7]는 양자택일의 관점을 탈피하여, 서명 검증 알고리즘에서 입력 값으로 받은 전자서명이 어떤 수준의 확률로 유효한 서명인지를 출력하는 탄력적인 전자서명 기법을 제시하였으며, 머클 트리(Merkle tree)[12] 기반의 Lamport-Diffie 일회용 전자서명 기법[13]을 이용하여 구체적인 예시를 제공하였다. 이 후, Taleb과 Vergnaud[8]는 Le 등의 기법을 확장하여 RSA 전자서명, 타원곡선 전자서명, GPV 전자서명 등에 적용하였다. 그러나 상술한 기법들은 전자서명의 확률적인 검증에 주목하여 연구를 진행하였으며, 본 논문에서 다루고 있는 사전계산을 활용한 양자택일 기반의 효율적인 검증과는 다소 거리가 있다.

오프라인상의 사전계산을 이용한 양자택일 기반의 효율적인 검증과 관련한 연구로는 Sipasseuth, Plantard, Susilo의 결과[14]를 들 수 있다. Sipasseuth 등은 주어진 행렬 A, B, C에 대하여 $AB=C$임을 확인하는 확률적이고 효율적인 방법인 Freivalds 알고리즘[15]을 이용하여 격자 기반의 DRS 전자서명[16]을 효

율적으로 검증하는 방법을 제시하였다. 그러나 Sipasseuth 등의 결과는 DRS 전자서명에만 국한되어 적용하였으며, 효율적인 검증보다는 DRS 전자서명의 검증에 필요한 메모리의 크기를 줄이는 것에 초점을 맞추고 있다. Boschini 등[10]은 Sipasseuth 등의 연구 결과를 확장하여 전자서명의 검증 알고리즘이 행렬과 벡터의 곱의 결과를 확인하는 형태로 주어지는 경우 사전계산을 활용하여 효율적으로 검증할 수 있도록 변환하는 일반적인 방법인 BFP 변환을 제시하였으며, 격자기반의 GPV 전자서명과 다변수 다항식 기반의 Rainbow 전자서명에 적용하여 구체적인 예를 보여주었다. 본 논문에서는 Boschini 등의 논문에서 고려하지 않았던 부호 기반 전자서명 분야에 대하여 BFP 변환을 적용하고 이에 관한 구현 결과에 대하여 다룬다.

3.2 전자서명의 효율적인 검증을 위한 BFP 변환

이번 절에서는 오프라인 사전계산을 이용하여 효율적인 전자서명 검증을 지원하는 최근 제시된 BFP 변환 방법[10]을 소개한다. BFP 변환 방법은 전자서명의 검증 알고리즘이 행렬과 벡터의 곱의 결과를 확인하는 형태로 이루어진 경우 해당 전자서명에 대한 효율적인 검증 방법을 제공하는 일반적인 변환 방법이다. 본 절에서는 구체적인 BFP 변환 방법을 살펴본다.

전자서명의 공개키가 $m \times n$행렬 M, 서명이 n차원 벡터 v, 메시지에 대한 해시함수가 m차원 벡터 u의 형태로 주어지는 전자서명에 대하여 검증 알고리즘에서

$$Mv = u \text{ in } F_q \tag{3}$$

가 성립하는지 확인한다고 하자. (이 때, q는 전자서명이 정의된 공간에 따라 결정되는 파라미터이다.) 식 (3)이 성립하는 경우, 임의의 m차원 벡터 c를 선택하여

$$c \bullet (Mv) = c \bullet u \quad \text{in } F_q \tag{4}$$

를 계산하면 이는 항상 성립하게 된다. (본 논문에서 두 벡터 a, b에 대하여 $a \bullet b$는 두 벡터의 내적 값으로 정의된다.) 나아가 식 (4)의 좌변은 $z = cM$, 우변은 $w = c \bullet u$로 표현할 경우,

$$z \bullet v = w \text{ in } F_q \tag{5}$$

로 볼 수 있다. 따라서, 식 (3)의 관계의 확인은 식 (5)를 확인하는 것으로 변환할 수 있다.

이때, 식 (3)에서 좌변을 계산하기 위해서는 약 mn번의 곱셈이 필요하지만, 식 (5)에서는 $z = cM$을 약 mn번의 사전계산을 통하여 미리 계산해 두면, 온라인 검증 과정에서는 $z \bullet v$와 $w = c \bullet u$를 위한 $m+n$번의 곱셈만으로 확인할 수 있다.

한편, 식 (5)의 경우 F_q에서 성립하는지 확인하게 됨에 따라, 식 (3)이 성립하지 않더라도 임의로 선택된 c의 값에 따라 식 (5)가 만족하는 경우가 발생할 수 있다. 이러한 경우가 발생할 확률은 $1/q$이고, 이 확률을 낮추기 위해서는 여러 개의 c를 선택하여 각각의 c에 대하여 식 (5)가 모두 성립하는지 확인한다. 실제 아래 기술하는 BFP 변환 알고리즘에서는 각각의 c가 독립적으로 랜덤하게 선택되어 식 (5)를 만족시킬 확률이 $1/q$라고 가정하여 파라미터를 선택한다. 즉, $1/q^l$의 확률로 l개의 c에 대하여 모두 식을 만족시킬 수 있다고 보며, 이는 유효하지 않은 서명을 유효한 서명으로 판단하는 잘못된 검증 결과를 출력하는 상황을 야기하게 된다. 아래 서술하는 알고리즘에서는 안전성 파라미터 λ에 대하여

$$1/q^l < 2^{-\lambda} \tag{6}$$

을 만족하는 정수 l을 선택하여 잘못된 검증 결과가 도출할 확률이 안전성 파라미터에 대하여 무시해도 될 만한(negligible) 수준이 되도록 한다. 또한, 실제 사용하는 시나리오에 따라 검증 알고리즘의 효율성을 고려하여 잘못된 검증 결과가 나올 확률이 다소 높도록 작은 파라미터 l을 설정하여 활용할 수도 있는데, l값에 따른 알고리즘의 효율에 관한 실험 결과는 4.4.3절에서 제공한다.

지금까지 살펴본 F_q 상에서 $Mv = u$를 만족하는 검증 알고리즘을 갖는 전자서명에 대하여 효율적인 검증을 위한 BFP 변환 방법을 정리하면 다음과 같다. M을 $m \times n$행렬, v를 n차원 벡터, u를 m차원 벡터라 하자.

1) 오프라인 단계(Offline Phase):
   (1) 안전성 파라미터 λ에 대하여 $1/q^l < 2^{-\lambda}$를 만족하는 정수 파라미터 l을 정한다.
   (2) l개의 m차원 벡터 c_i, $i = 1, \cdots, l$을 임의로 선택한다.
   (3) 모든 $i = 1, \cdots, l$에 대하여 $z_i = c_i M$을 계산하여, (c_i, z_i)를 출력한다.

2) 온라인 단계(Online Phase): 서명에 대한 벡터 v와 메시지에 대한 벡터 u가 주어지면,
   (1) 모든 $i=1, \cdots, l$에 대하여 $z_i \bullet v = c_i \bullet u$ in F_q 가 성립하는지 확인한다.
   (2) 모든 i에 대하여 성립하면 1(Accept)을 출력하며, 그렇지 않으면 0(Reject)을 출력한다.

검증 과정에서 위와 같이 변환을 적용하여도 기존 전자서명의 안전성과 비교하여 같은 안전성을 유지할 수 있다. 실제로 오프라인 단계에서 설명한 알고리즘은 임의의 벡터 c_i와 공개키인 행렬 M에 관한 결과로만 연산이 수행된다. 또한, 오프라인과 온라인 단계 모두 검증자가 계산하여 결과를 갖기 때문에 $z_i = c_i M$ 값은 서명을 생성하는 쪽에서는 비밀 값이 된다. 따라서, 관련 정보가 없으므로 서명을 생성하는 사람이 변환된 알고리즘의 온라인 단계를 통과하도록 서명을 위조하여 생성하는 것은 기존의 서명 알고리즘과 마찬가지로 무시할만한 수준의 확률을 제외하고는 불가능하다.

위의 과정을 계산량 관점에서 살펴보면 오프라인 단계의 계산량은 mnl의 곱셈이 필요하며, 온라인 단계에서는 $(m+n)l$의 곱셈이 필요하다. 기존 $Mv=u$의 검증을 위해서는 mn번의 계산이 필요한데, 일반적으로 $2l \ll m$이고 $2l \ll n$이므로, 오프라인 단계의 연산을 희생하여 온라인 단계의 계산 시간이 크게 줄어드는 효과를 얻을 수 있다. 온라인과 오프라인 계산을 통합하여 효율성을 살펴보면, 서명을 t번 검증한다고 할 때, 기존의 경우 mnt번의 계산이 필요하지만, BFP 변환을 적용하면 mnl번의 오프라인 계산과 $(m+n)lt$의 온라인 계산이 필요하게 된다. 이를 정리하면, 전체 계산량 측면에서 BFP 변환이 더 효율적이기 위해서는 $mnt > mnl + (m+n)lt$을 만족시켜야 하며, 이는

$$t > \frac{mnl}{mn-(m+n)l} \tag{7}$$

와 같게 된다. Wave 전자서명의 구체적인 파라미터에 대한 BFP 변환의 효율은 4장에서 다시 살펴보도록 한다.

저장 공간 관점에서 기존에는 행렬 M을 저장해야 하므로, mn개의 F_q의 원소를 위한 저장 공간이 필요하지만, 위의 방법으로는 l개의 m차원 벡터 c_i와 n차원 벡터 z_i가 필요하므로, $l(m+n)$개의 F_q의 원소를 저장해야 하므로 필요한 저장 공간 또한 줄어들게 된다.

3.2.1 기존 논문에서의 BFP 변환과 본 논문의 서술적 차이

앞서 제시한 식 (4)에서 우변의 결과를 좌변으로 이항하여

$$c \bullet (Mv) - c \bullet u = 0 \ \text{ in } \ F_q$$

로 쓸 수 있고, 이는

$$c[M \mid -I]\begin{bmatrix} v \\ u \end{bmatrix} = 0 \ \text{ in } \ F_q \tag{8}$$

와 같은 형태로 나타낼 수 있다. (이 때, $[M \mid -I]$는 $m \times n$ 행렬 M의 오른쪽에 M과 행의 개수가 같은 $m \times m$ 항등행렬 I에 대하여 $-I$를 붙여 쓴 $m \times (m+n)$ 행렬을 나타내며, $\begin{bmatrix} v \\ u \end{bmatrix}$는 길이가 n인 벡터 v의 아래쪽에 길이가 m인 벡터 u를 붙여 쓴 길이가 $n+m$인 벡터를 의미한다.) 이는 앞에서 제시한 것과 같이 $z = cM$ 대신 $z = c[M \mid -I]$로 놓으면

$$z\begin{bmatrix} v \\ u \end{bmatrix} = 0 \ \text{ in } \ F_q \tag{9}$$

임을 확인하는 것과 같게 된다. z를 계산하기 위해서는 $m(m+n)$번의 곱셈이 필요하며, 식 (4)의 경우 mn번의 곱셈이 필요하므로 기존 논문에서의 오프라인 계산량이 본 논문에서 서술한 것보다 많다. 온라인의 경우 식 (9)는 $m+n$의 곱셈이 필요하여 식 (5)를 확인하는 것과 동일한 계산량이 필요하다. 또한, 실제 구현하는 과정에서는 기존에 공개키 및 서명으로 주어진 값 M, u, v에 대하여 $[M \mid -I]$와 $\begin{bmatrix} v \\ u \end{bmatrix}$로 변환하는 과정이 필요하게 되어 추가적으로 시간을 소요하게 된다. 이에 본 논문에서는 식 (8)과 식 (9) 대신 식 (4)와 식 (5)로 각각 구현을 진행하였다. 이에 관한 실험 결과는 4.4.4절에서 비교하도록 한다.

Ⅳ. Wave 전자서명의 효율적인 검증 적용 및 실험 결과

이번 장에서는 Wave 전자서명에 BFP 변환 방법을

적용해보며, 구현을 통하여 이와 관련된 여러 실험 결과를 살펴보도록 한다.

4.1 Wave 전자서명의 효율적인 검증

2.3절에서 살펴본 바와 같이, Wave 전자서명의 검증 알고리즘 Wave.Verify는 공개키 pk와 서명 $\sigma=(eP,r)$을 입력 값으로 받아 검증하고자 하는 메시지 m과 해시함수 $Hash$에 대하여 $s'=Hash(m,r)$을 계산한 뒤, e'의 해밍 웨이트가 w인지 확인하고, $e'(H_{pk})^T=s'$이 성립하는지 확인한다. 이 중, 식 $e'(H_{pk})^T=s'$은 양변에 전치행렬의 연산을 취하면, $H_{pk}e'^T=s'^T$와 동치임을 알 수 있으며, 이는 H_{pk}가 공개키와 연관된 행렬인 M, e'^T가 서명과 연관된 벡터인 v, s'^T가 메시지와 연관된 벡터인 u에 대응되는 $Mv=u$의 형태임을 확인할 수 있다. 따라서, Wave 전자서명에도 효율적인 검증 알고리즘을 위한 BFP 변환을 적용할 수 있다.

2장에서 소개한 Wave 전자서명의 파라미터에 따르면, 이용한 $(U,U+V)$-부호는 길이가 n인 벡터들의 공간에서 k차원의 부호로 정의가 되었다. 따라서 H_{pk}는 $(n-k)\times n$행렬이며, e'은 n차원 벡터, s'은 $(n-k)$차원의 벡터이다. 여기에 3장의 분석 결과를 적용하면, 오프라인 단계의 계산량은 $(n-k)nl$의 곱셈이 필요하며, 온라인 단계에서는 계산량이 $(2n-k)l$이다. 반면, Wave의 검증을 위해서는 $(n-k)n$의 곱셈이 필요하다. 또한 저장해야하는 값의 경우에도 기존의 Wave 전자서명의 경우 공개키의 저장 또는 교환을 위하여 $(n-k)n$개의 F_q의 원소를 저장해야 하나, BFP 변환의 경우 $(2n-k)l$개의 F_q 원소를 저장해야 한다. Wave의 구체적인 파라미터에 대해서는 다음 절에서 살펴보도록 한다.

4.2 실험 환경 및 Wave 전자서명 파라미터

본 논문에서 제시된 결과에 대한 실험 환경은 <표 1>과 같다. 실험에 사용한 PC는 Intel(R) Core(TM) i7-11700@2.50GHz의 CPU에 32GB RAM을 갖추었다. OS는 Windows 10에서 제공하는 WSL(Windows Subsystem for Linux)을 활용하여 Linux인 Ubuntu 20.04 LTS에서 실험하였다.

표 1. 실험 환경
Table 1. Experimental environment

Item	Specification
CPU	Intel(R) Core(TM) i7-11700@2.50GHz
RAM	32GB
OS	Windows 10 WSL-Ubuntu 20.04 LTS

Wave 부호기반 전자서명의 파라미터는 안전성 강도에 따라 64비트, 80비트, 96비트, 128비트의 4가지로 구성되어 있다. 기존의 논문[5]에서 제안된 바와 같이 $q=3$으로 고정하여 사용하였으며, 각각의 안전성 강도에 따른 파라미터의 크기는 표 2와 같이 설정하였다.

표 2. 실험에 사용한 Wave 전자서명의 파라미터
Table 2. Parameters of Wave signatures for our implementation

Security	64 bits	80 bits	96 bits	128 bits
block length (n)	4,246	5,308	6,368	8,492
error weight (w)	3,990	4,988	5,984	7,890
dimension (k)	2,803	3,504	4,203	5,605
dimension of U (k_U)	1,779	2,224	2,668	3,558
dimension of V (k_V)	1,024	1,280	1,535	2,047

4.3 Wave 부호 기반 전자서명의 수행 시간

기존의 Wave 전자서명의 구현을 위하여 논문의 저자들이 제시한 소스코드[17]를 활용하였다. 표 3은 본 논문의 실험 환경에서 수행한 Wave 부호기반 전자서명의 키 생성, 서명 생성, 검증 알고리즘을 보여준다. 표 3의 데이터는 각 알고리즘을 1,000번 수행한 평균값이다. 주어진 알고리즘 수행 시간에 대한 데이터의 단위는 ms이다. 실험 결과에 따르면, 128비트 안전성을 달성하는 파라미터 하에서 Wave 전자서명은 키 생성에 1,369ms, 서명 생성에 322ms가 필요하며, 서명의 검증에는 1.107ms가 소요된다. 그림 1, 2, 3에서는 Wave 전자서명의 키 생성, 서명 생성, 검증 시간을 각각 그래프로 보여준다.

표 3. Wave 전자서명의 키 생성과 서명 생성, 검증 알고리즘 수행 시간
Table 3. Running time of key generation, sign, and verification algorithms of Wave

Security	64 bits	80 bits	96 bits	128 bits
KeyGen (Unit: ms)	270	444	697	1,369
Sign (Unit: ms)	59	102	160	322
Verify (Unit: ms)	0.296	0.436	0.654	1.107

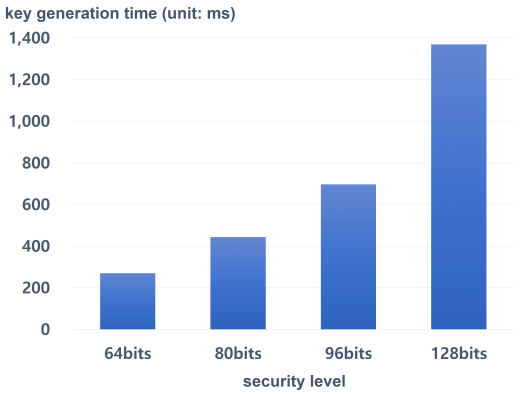

그림 1. 안전성 파라미터에 따른 Wave 전자서명의 키 생성 시간
Fig. 1. Key generation time of Wave with respect to security parameters

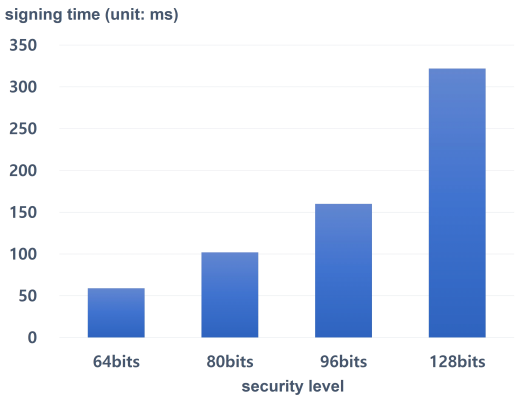

그림 2. 안전성 파라미터에 따른 Wave 전자서명의 서명 생성 시간
Fig. 2. Signing time of Wave with respect to security parameters

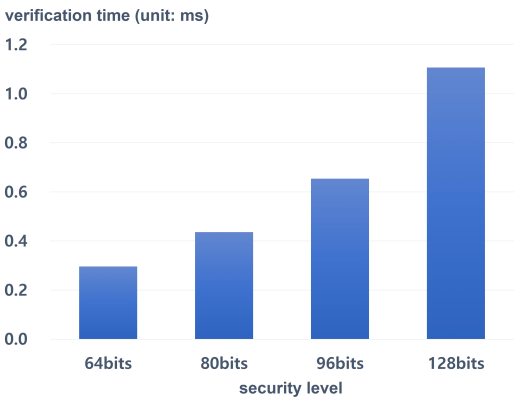

그림 3. 안전성 파라미터에 따른 Wave 전자서명의 검증 시간
Fig. 3. Verification time of Wave with respect to security parameters

4.4 BFP 변환 적용 Wave 전자서명의 구현

이번 절에서는 Wave 전자서명에 대하여 BFP 변환을 적용한 여러 실험 결과에 대하여 살펴본다.

4.4.1 BFP 변환 적용 Wave 전자서명 구현 결과

표 4는 BFP 변환을 적용한 Wave 전자서명에 대하여 안전성 파라미터 λ에 따라 식 (6)을 만족하기 위하여 필요한 최소의 c_i들의 개수 l, 오프라인 단계 수행 시간, 유효한 서명과 유효하지 않은 서명 각각에 대한 검증 시간을 보여준다. 각각의 안전성 파라미터에 대하여 100번의 오프라인 단계를 수행하였으며, 생성된 오프라인 단계의 결과마다 1,000번의 서명 검증을 수행하여 얻은 평균값을 표의 데이터로 제시한다.

3.2절에서 설명한 바와 같이, 식 (3)이 성립하지 않는 경우에도 임의의 벡터 c에 대하여 식 (5)를 만족시킬 확률이 $1/q$가 된다. 따라서, q의 값이 작은 경우 정확도를 높이기 위해서는 여러 개의 랜덤하게 선택된 c벡터에 대하여 식 (5)를 확인해야 한다. 본 실험의 경우 $q=3$으로 작은 값이어서 여러 개의 c벡터를 선택하여 정확도를 높여야 하며, 안전성 파라미터 λ에 대하여 잘못된 결과가 나올 확률이 $2^{-\lambda}$보다 작게 되는 값 중 가장 작은 정수 값을 l로 설정하였다. 즉, l은

$$3^{-l} < 2^{-\lambda} \tag{10}$$

를 만족하는 가장 작은 정수로 선택하였다. 이 때, 각 c에 대하여 식 (5)를 만족시키는 사건은 독립적인 것으로 가정하였으며, l의 개수가 작을수록 높은 효율을 얻을 수 있으므로 효율성을 높이기 위하여 위의 식을 만족하는 가장 작은 정수 값을 l로 설정하였다.

표 4. BFP 변환을 적용한 Wave 알고리즘의 수행 시간
Table 4. Running time of algorithms for Wave with BFP transformation

Security	64 bits	80 bits	96 bits	128 bits
The number of $c_i's$ (l)	41	51	61	81
Offline Time (Unit: ms)	845	1,638	2,788	6,653
Online Time (valid signatures/ Unit: ms)	0.035	0.046	0.058	0.091
Online Time (invalid signatures/ Unit: ms)	0.021	0.026	0.031	0.042

그림 4는 BFP 기법을 적용한 Wave 전자서명의 안전성 파라미터에 따른 오프라인 단계 소요 시간을 그래프로 보여준다. 실험 결과에 따르면, 64비트, 80비트, 96비트, 128비트 안전성을 달성하는 Wave의 파라미터에 대하여 각각 845ms, 1,638ms, 2,788ms, 6,653ms의 오프라인 단계 수행 시간이 필요하다. 4.1절의 분석에 따르면, 오프라인 단계에서 필요한 곱셈의 이론적 계산량은 $(n-k)nl$로 이 값은 64비트 안전성을 만족하는 경우에 비하여 80비트 안전성의 경우 1.94배, 96비트 3.35배, 128비트 7.91배 증가하게 되며, 이는 실험결과와 유사한 결과이다. 제시한 실험결과에 따르면, 오프라인 단계에서 많은 시간을 소요함을 확인할 수 있다. 오프라인 단계의 경우 온라인 단계가 시작되기 전에 한 번만 계산해 두면, 이를 이용하여 많은 서명을 검증할 수 있으므로 동일한 서명자에 대한 서명을 빈번히 검증해야 하는 경우 큰 문제가 되지 않을 수 있다. 반대로 검증해야 하는 서명의 수가 많지 않다면, 안전성 파라미터를 고려하지 않고 유효하지 않은 서명이 검증을 통과할 확률을 적절히 크게 설정하여 l값을 작게 할 수 있으며 이를 이용하여 오프라인 단계의 소요 시간을 줄여 효율을 높일 수도 있다. 이와 관련하여 4.4.3절에서 자세히 살펴보도록 한다.

온라인 단계에서는 식 (5)를 만족시키지 않는 c가 등장하면 바로 작업을 멈추고, 0(Reject)을 출력하게 된다. 따라서, 효율적인 검증 알고리즘에서는 유효하지 않은 서명의 검증이 유효한 서명의 검증보다 평균적으로 적은 시간이 걸리게 된다. 이를 확인하기 위하여, 유효한 서명과 유효하지 않은 서명을 구분하여 온라인 검증 시간을 측정하였다. 그림 5는 유효한 서명과 유효하지 않은 서명에 대하여 안전성 파라미터에 따른 BFP 변환을 적용한 Wave 전자서명의 온라인 단계 소요 시간을 그래프로 보여준다. 결과에 따르면, 128비트의 안전성을 달성하는 파라미터에서 유효한 서명을 검증하기 위해 0.091ms, 유효하지 않은 서명을 검증하기 위해 0.042ms의 시간이 소요되어 유효한 서명을 검증하기 위하여 2.17배 이상의 시간이 더 소요됨을 확인하였다. 이와 관련된 구체적인 분석은 다음 절에서 자세히 살펴본다.

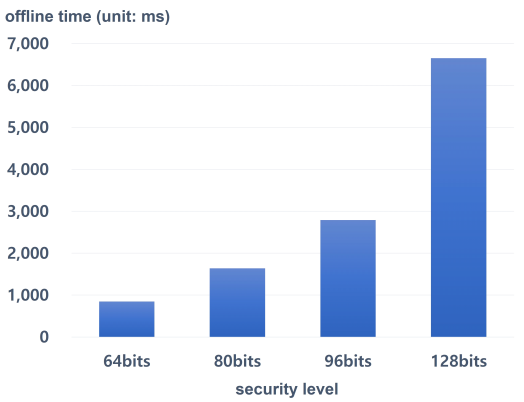

그림 4. BFP 기법을 적용한 Wave 전자서명의 오프라인 단계 소요 시간
Fig. 4. Running time of the offline phase for Wave with BFP transformation

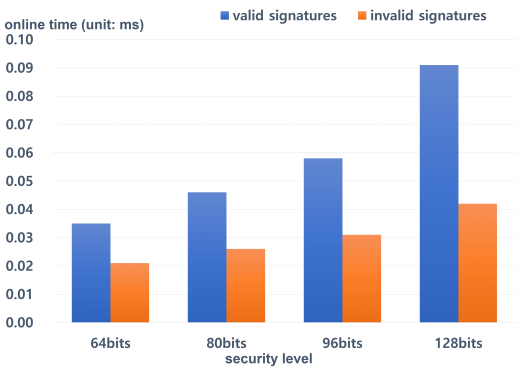

그림 5. BFP 변환을 적용한 Wave 전자서명의 온라인 단계 소요 시간
Fig. 5. Running time of the online phase for Wave with BFP transformation

표 5는 기존의 Wave 전자서명과 효율적인 검증을 위한 기법을 적용한 Wave 전자서명의 검증과정 중 온라인 단계 소요 시간을 비교한다. 분석 결과에 따르면, 128비트 안전성을 갖는 파라미터 하에서 효율적인 검증을 위한 기법을 적용하는 경우, 기존의 전자서명에 비하여 12배가량 빨라지는 것을 확인할 수 있다.

표 5. 기존의 Wave 전자서명과 BFP 변환을 적용한 전자서명의 온라인 시간 비교
Table 5. Comparison of online times between the original Wave and Wave with BFP transformation

Security	64 bits	80 bits	96 bits	128 bits
Original Wave (A) (Unit: ms)	0.296	0.436	0.654	1.107
Wave with BFP transformation (B) (Unit: ms)	0.035	0.046	0.058	0.091
(A)/(B)	8.45	9.48	11.28	12.16

4.4.2 서명의 유효성에 따른 온라인 단계의 분석

이전 절에서 언급한 바와 같이 유효하지 않은 서명의 경우 식 (5)를 만족시키지 않는 c가 등장하면 바로 작업을 멈추고 0(Reject)을 출력하기 때문에, 유효한 서명의 검증보다 유효하지 않은 서명의 검증에 더 적은 시간을 소요하게 된다. 이때, 유효하지 않은 서명에 대하여 임의의 c가 식 (5)를 만족시킬 확률은 $1/q$라고 가정하였으며, 본 논문에서 구현한 Wave 전자서명의 경우 $q=3$이므로 $1/3$의 확률로 식 (5)를 만족시키게 된다. 또한, 각각의 c에 대하여 식 (5)를 만족시키는 사건은 독립이라고 가정하자. 그러면, 식 (5)를 만족시키지 않는 c값을 얻게 되는 기댓값은

$$\sum_{k=1}^{\infty} k\left(\frac{1}{3}\right)^{k-1}\frac{2}{3}$$

이고, 이 값은 1.5가 된다. 실제로 구현 결과, 유효하지 않은 서명에 대하여 각각의 안전성 파라미터에서 평균적으로 1.495~1.505개의 c에 대해 식 (5)를 확인하여 유효하지 않은 서명임을 확인할 수 있었다. 반대로 이 결과는 각각의 c에 대하여 식 (5)를 만족시키는 사건이 독립적이라는 가정이 적절한 가정이라는 것을 보여주며, 이는 3.2절에서 식 (6)을 얻기 위한 가정도 적절하였음을 간접적으로 확인할 수 있다.

위에서 제시한 결과에 따르면, 유효하지 않은 서명은 약 1.5개의 c에 대하여 식 (5)를 검증하면 되지만, 유효하지 않은 서명의 경우 l개의 c에 대하여 검증하게 되므로 유효하지 않은 서명이 유효한 서명에 비하여 약 $1.5/l$배의 검증 시간이 필요할 것으로 보인다. 그러나 실제 구현 결과는 이론값보다 적은 차이를 보여주었는데, 이는 메시지 해시 값의 자료형 변환 등 온라인 단계를 수행하는 데 필요한 보조 계산의 영향 때문이다. 표 6은 온라인 단계에서 유효한 서명과 유효하지 않은 서명에 대하여 식 (5)를 확인하는 Step 1과 나머지 부분에 대한 소요 시간을 보여준다. 이에 따르면, Step 1의 소요 시간의 경우 유효하지 않은 서명이 유효한 서명에 비하여 약 $1.5/l$배만큼 적은 것을 알 수 있지만, 부가적인 계산 시간이 같게 필요하여 전체 온라인 단계의 소요 시간에서는 이보다 적은 비율의 차이가 나는 것을 확인할 수 있다.

표 6. 유효한 서명과 유효하지 않은 서명의 온라인 단계의 계산별 소요 시간
Table 6. Running time for steps of the online phase with respect to validity of signatures

Security	Valid signatures		Invalid signatures	
	Step 1 (Unit: ms)	Auxiliary Steps (Unit: ms)	Step 1 (Unit: ms)	Auxiliary Steps (Unit: ms)
64 bits	0.015	0.020	0.001	0.020
80 bits	0.021	0.025	0.001	0.024
90 bits	0.029	0.029	0.002	0.029
128 bits	0.052	0.039	0.002	0.039

4.4.3 벡터의 개수 l값에 따른 구현 결과

4.4.1절에서 살펴본 바와 같이 Wave 전자서명에 BFP 변환을 적용하는 경우, 오프라인 단계에서 많은 시간을 소요하는 것을 확인하였다. 3.3절에서 제시한 식 (7)의 이론적인 분석을 4.4.1절에서 얻은 실험 결과로 대체하면, 오프라인 단계를 활용하는 점을 고려하지 않고 전체 소요 시간을 그대로 비교할 때 기존의 Wave 전자서명보다 효율적으로 검증하기 위해서는 t개의 서명에 대하여

$$\begin{aligned}&(\text{오프라인 소요시간})+t\times(\text{온라인 소요시간})\\ &< t\times(\text{Wave 전자서명의 검증 시간})\end{aligned} \tag{11}$$

을 만족해야 한다. 각 안전성 파라미터에 대하여 위의 조건을 만족시키는 t의 값은 표 7과 같다.

표 7에 따르면, BFP 변환을 적용한 Wave의 경우에 많은 수의 서명을 검증할 때 기존의 Wave 서명보다 효율적임을 확인할 수 있다. 그러나 이 결과는 식 (10)을 만족시키는 파라미터에 관한 결과로 안전성 파라미터에 따라 각각 2^{64}개, 2^{80}개, 2^{96}개, 2^{128}개의 유효하지 않은 서명을 검증하는 동안 유효하다고 잘못 결론을 내는 경우의 기대값이 1개 이하가 되도록 파라미터를 설정한 결과이다. 만약, 파라미터를 여유 있게 설정하여 기존보다 적은 수의 유효하지 않은 서명을 검증하는 동안 잘못된 결론을 내지 않도록 파라미터를 설정하면 l값을 기존보다 작은 값으로 설정할 수 있고, 이는 알고리즘의 효율성을 높여주는 결과를 가져다줄 수 있다.

표 7. 안전성 파라미터에 따른 식 (11)을 만족시키는 서명의 개수
Table 7. The number of signatures required for satisfying Eq. (11)

Security	t
64 bits	3,238
80 bits	4,201
96 bits	4,678
128 bits	6,548

l값에 따른 알고리즘의 효율성을 살펴보기에 앞서, l값에 따른 유효하지 않은 서명이 유효한 서명으로 판단될 확률을 살펴본다. 3.2절의 식 (6)과 4.4.1절의 식 (10)에서 이론적으로 분석한 것과 같이 임의의 c에 대하여 식 (5)를 만족시키는 사건이 각각의 c에 대하여 독립적이라고 가정하면, 유효하지 않은 서명이 주어졌을 때 l개의 c에 대하여 모두 식 (5)를 만족시킬 확률은 $1/q^l = 1/3^l$이다. 따라서, 온라인 단계에서 l개의 c에 대하여 식 (5)를 모두 만족시키는 경우 검증에 성공한다고 할 때, s개의 유효하지 않은 서명 중 유효한 서명으로 판단되는 서명 개수의 기댓값은 $s/q^l = s/3^l$이 된다.

표 8은 128비트 안전성을 갖는 파라미터에서 l값에 따른 100,000개의 유효하지 않은 서명 중 유효한 서명으로 판단되는 서명의 수에 대한 실험 결과를 보여준다. 실험 결과에 따르면, l이 증가함에 따라 이론적인 분석 결과와 같은 경향을 보이는 것을 확인할 수 있다.

표 9는 각각의 안전성 파라미터에 대하여 l의 개수에 따른 오프라인과 온라인 단계의 소요 시간을 보여준다. 실험 결과에 따르면, 오프라인 시간은 l값에 비례함을 확인할 수 있으며, 온라인 시간의 경우 l값에 정확히 비례하지는 않으나 l에 따라 일정한 간격을 유지함을 확인할 수 있다. 이는 4.4.2절의 표 6에서 살펴본 바와 같이 온라인 단계를 수행하는 데 필요한 보조 계산의 소요 시간이 영향을 주기 때문으로 보인다. 표 9의 결과에 따르면, l이 10 이상인 경우 t값이 2^l보다 작은 값이 되어 가장 작은 t개의 유효하지 않은 서명을 검증하는 동안 평균적으로 1개 이하의 서명이 유효한 서명으로 잘못 판단되게 된다. 따라서 상대적으로 작은 l을 사용하여도 변환된 서명 기법이 올바르고 효율적으로 동작할 수 있음을 확인할 수 있다.

표 8. l값에 따른 100,000개의 유효하지 않은 서명 중 유효한 서명으로 잘못 판단하는 비율
Table 8. The ratio of wrong verification results on 100,000 invalid signatures with respect to l

l	The number of wrong verification results	Ratio (%)
2	10,935	10.935
4	1,219	1.219
6	147	0.147
8	12	0.012
10	3	0.003
12	0	0

표 9. l 값에 따른 BFP 변환을 적용한 Wave 전자서명의 알고리즘 소요 시간
Table 9. Running time of Wave with BFP transformation for various l

64 bits				
l	10	20	30	40
Offline Time (Unit: ms)	223	424	624	819
Online Time (Unit: ms)	0.024	0.028	0.031	0.035
The smallest t satisfying Eq. (11)	820	1,582	2,355	3,138
80 bits				
l	10	20	30	40
Offline Time (Unit: ms)	351	657	970	1,283
Online Time (Unit: ms)	0.030	0.034	0.038	0.042
The smallest t satisfying Eq. (11)	865	1,635	2,438	3,257
96 bits				
l	10	20	30	40
Offline Time (Unit: ms)	521	947	1,376	1,820
Online Time (Unit: ms)	0.036	0.040	0.044	0.049
The smallest t satisfying Eq. (11)	844	1,543	2,256	3,009
128 bits				
l	10	20	30	40
Offline Time (Unit: ms)	942	1699	2500	3348
Online Time (Unit: ms)	0.046	0.051	0.058	0.067
The smallest t satisfying Eq. (11)	888	1,609	2,384	3,220

4.4.4 기존 논문의 알고리즘과 본 논문의 변형 알고리즘에 대한 실험 결과 비교

본 논문에서 제시한 것과 달리 기존의 BFP 변환 기법은 3.2.1절에서 소개한대로 식 (4)와 식 (5) 대신 식 (8)과 식 (9)로 변경하여 제시하였다. 이번 절에서는 식 (8)과 식 (9)의 형태로 구현을 하는 경우의 실험 결과를 제시하고, 식 (4)와 식 (5)의 형태로 구현하는 경우와 비교하고자 한다.

표 10은 Wave 전자서명에 기존의 BFP 변환을 적용하는 경우와 본 논문에서 제시한 변형을 적용하는 경우에 따른 실험 결과를 제공한다. 실험 결과에 따르면, 본 논문에서 제시한 변형을 제공하는 경우 이론적인 분석에서 차이를 보였던 것과 같이 오프라인 단계

표 10. 기존의 BFP 변환과 본 논문의 변형의 알고리즘 소요 시간 분석
Table 10. Comparison of running time between the original BFP and our modification

Security		64 bits	80 bits	96 bits	128 bits
The number of c_i's (l)		41	51	61	81
Original BFP	Offline Time (Unit: ms)	927	1,800	3,071	7,372
	Online Time (valid signatures/ Unit: ms)	0.033	0.042	0.055	0.085
	Online Time (invalid signatures/ Unit: ms)	0.021	0.026	0.031	0.041
Ours	Offline Time (Unit: ms)	845	1,638	2,788	6,653
	Online Time (valid signatures/ Unit: ms)	0.035	0.046	0.058	0.091
	Online Time (invalid signatures/ Unit: ms)	0.021	0.026	0.031	0.042

의 시간을 약 10% 줄일 수 있었다. 반면, 온라인 단계 계산의 경우 이론적으로는 같은 계산량을 가지나, 실제 계산 과정에서 기존의 BFP 변환의 경우 1번의 벡터 내적 연산 호출로 계산이 되지만, 본 논문에서 제시한 변형은 2번의 벡터 내적 연산을 호출하여 6-9% 가량의 시간을 더 사용하는 것으로 측정되었다. 그 결과, 오프라인과 온라인 단계를 더한 전체 소요 시간에서 본 논문에서 제시한 것이 더 효율적임을 확인할 수 있다.

V. 결 론

본 논문에서는 최근 제시된 대표적인 부호 기반 전자서명인 Wave 전자서명의 효율적인 검증 방법을 제시한다. 최근 Boschini, Fiore, Pagnin은 전자서명의 검증 알고리즘이 특정 형태를 만족하면 효율적인 검증 알고리즘으로 변환하는 BFP 변환 방법을 제시하였다. 본 논문에서는 Wave 전자서명의 검증 알고리즘이 BFP 변환을 적용하기 위한 특정 형태를 만족함을 확인하였다. 나아가 구현을 통하여 제시한 Wave 전자서명의 효율적인 검증 방법의 성능 향상 정도를 확인하였다. 본 논문에서 제시한 구현 결과에 따르면, 128 비트 안전성을 만족하는 Wave 전자서명의 경우, 약 6,653ms의 오프라인 계산을 해두면 이 값을 이용하여 서명의 온라인 검증을 0.091ms의 시간에 수행할 수 있는데, 이는 기존의 Wave 서명의 온라인 검증 시간을 약 12배 향상시키는 결과이다.

References

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120-126, Feb. 1978.

[2] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. ACM STOC 2008*, pp. 197-206, May 2008.

[3] J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in *Proc. ACNS 2005*, pp. 164-175, Jun. 2005.

[4] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe, "The SPHINCS+ signature framework," in *Proc. ACM CCS 2019*, pp. 2129-2146, Nov. 2019.

[5] T. Debris-Alazard, N. Sendrier, and J. P. Tillich, "Wave: A new family of trapdoor one-way preimage sampleable functions based on codes," in *Proc. ASIACRYPT 2019*, pp. 21-51, Dec. 2019.

[6] W. Lee, J. No, and Y. Kim, "RM code-based signature scheme," in *Proc. KICS ICC 2017*, pp. 1624-1625, Jeju Island, Korea, Jun. 2017.

[7] D. V. Le, M. Kelkar, and A. Kate, "Flexible signatures: Making authentication suitable for real-time environments," in *Proc. ESORICS 2019*, pp. 173-193, Sep. 2019.

[8] A. R. Taleb and D. Vergnaud, "Speeding-up verification of digital signatures," *J. Comput. and Syst. Sci.*, vol. 116, no. 2, pp. 22-39, Mar. 2021.

[9] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36-63, Jan. 2001.

[10] C. Boschini, D. Fiore, and E. Pagnin, "Progressive and efficient verification for digital signatures," in *Proc. ACNS 2022*, pp. 440-458, Rome, Italy, Jun. 2022.

[11] N. T. Courtois, M. Finiasz, and N. Sendrier, "How to achieve a McEliece-based digital signature scheme," in *Proc. ASIACRYPT 2001*,

pp. 157-174, Gold Coast, Autralia, Dec. 2001.
[12] R. C. Merkle, “A digital signature based on a conventional encryption function,” in *Proc. CRYPTO 1987*, pp. 369-378, Santa Barbara, USA, Aug. 1987.
[13] L. Lamport, “*Constructing digital signatures from a one-way function,*” Technical Report-CSL-98, SRI International, 1979.
[14] A. Sipasseuth, T. Plantard, and W. Susilo, “Using Freivalds’ algorithm to accelerate lattice-based signature verifications,” in *Proc. ISPEC 2019*, pp. 401-412, Kuala Lumpur, Malaysia, Nov. 2019.
[15] R. Freivalds, “Fast probabilistic algorithms,” in *Proc. MFCS 1979*, pp. 57-69, Olomouc, Czechoslovakia, Sep. 1979.
[16] T. Plantard, A. Sipasseuth, C. Dumondelle, and W. Susilo, “DRS: Diagonal dominant reduction for lattice-based signature,” in *Proc. First PQC Stand. Conf.*, Florida, USA, Apr. 2018.
[17] D.-A. Thomas, N. Sendrier, and J.-P. Tillich, “Implementation of wave: A code-based digital signature scheme,” Retrieved Oct. 15, 2021, from http://wave.inria.fr/en/.

임 성 창 (Seongchang Im)

2015년 8월~현재 : 중소벤처기업진흥공단
2016년 2월 : 전북대학교 컴퓨터공학부 학사
2020년 9월~현재 : 전북대학교 컴퓨터공학부 석사과정
<관심분야> 정보보호, 전자서명, 컴퓨터네트워크
[ORCID:0000-0001-8341-4070]

최 성 봉 (Seongbong Choi)

2020년 2월 : 전북대학교 컴퓨터공학부 학사
2021년 9월~현재 : 중앙대학교 컴퓨터공학과 석사과정
<관심분야> 공개키 암호, 전자서명, 다자간 연산
[ORCID:0000-0001-5499-829X]

조 기 환 (Gihwan Cho)

1985년 : 전남대학교 계산통계학과 학사
1987년 : 서울대학교 계산통계학과 석사
1987년~1997년 : ETRI 컴퓨터연구단 선임연구원
1996년 : Univ. of Newcastle 전산학과 박사
1997년~1999년 : 목포대학교 컴퓨터과학과 전임강사
1999년~현재 : 전북대학교 컴퓨터공학부 교수
<관심분야> 이동컴퓨팅, 컴퓨터네트워크, 정보보호, 무선인터넷
[ORCID:0000-0003-4923-8565]

이 형 태 (Hyung Tae Lee)

2006년 2월 : 서울대학교 수리과학부 학사
2008년 2월 : 서울대학교 수리과학부 석사
2013년 2월 : 서울대학교 수리과학부 박사
2013년 3월~2014년 2월 : 서울대학교 자연과학대학 박사후 연구원
2014년 5월~2017년 8월 : 싱가포르 난양이공대학 Research Fellow
2017년 9월~2021년 8월 : 전북대학교 컴퓨터공학부 조교수
2021년 9월~현재 : 중앙대학교 소프트웨어학부 조교수
<관심분야> 정보보호, 암호학
[ORCID:0000-0002-0920-2026]