Constructing the Cryptographic Boundaries for Lattice-based Cryptography on Hardware Security Module

Junting Xiao[1,a)] Tadahiko Ito[1,b)]

Abstract: Post-Quantum Cryptography (PQC) is regarded as an effective way to resist attacks with quantum computers. Since National Institute of Standards and Technology (NIST) proposed its PQC standardization project in 2016, many candidates have been submitted and their quantum-resistant capability has been measuring by researchers. Besides these researches, it is also indispensable to evaluate the practicality of PQC on constrained environments such as Hardware security module (HSM), which is designed to provide a trusted environment to perform cryptographic operations. In this paper, we assume the cases of using HSM for key management, and discuss the practicality of applying lattice-based cryptographies, which is one of the candidates of NIST's PQC project on it. We focus on the efficiency of hash operations instead of asymmetric operations, with different constructions of cryptographic boundaries. Then we point out that the bottleneck of PQC operations can be hash operation instead of asymmetric operation. Especially for the cases of document signing and code signing, large files would be signed, and this bottleneck will affect their efficiency. We chose three lattice-based digital signature schemes from round 2 of NIST's PQC project. We also analyses the bottleneck of these schemes and compare their performances under different constructions of cryptographic boundary when applied to HSM. After that, we propose the appropriate way to construct cryptographic boundaries for lattice-based cryptographic schemes when applied to HSM. We believe that our result helps to define cryptographic boundary for PQC, where theoretical proof and clearance of patents should be done.

Keywords: Lattice-based cryptography, hardware security module, cryptographic boundary

1. Introduction

A hardware security module (HSM) is a physical computing device that provides a trusted environment to perform cryptographic operations such as encryption, decryption and authentication, etc. HSMs typically satisfy the FIPS 140-2 [12] and/or common criteria standard to achieve the high security. Relying on secure mechanisms to create an isolated environment from normal computing environments, HSMs ensure reliable generation, protection, and managment of keys and sensitive data. It is now widely used in some critical infrastructure, for instance, be used as a part of public key infrastructure (PKI) or internet bank infrastructure. In these cases, many HSMs are connected together to preserve high practicality and efficiency. In the other hand, Shor introduced an algorithm [20] to solve the integer factorization problem and discrete logarithms problem which can break RSA and elliptic curve cryptography (ECC) on quantum computers. In that case, HSMs which are still using traditional cryptography may not be able to protect and manage their keys and sensitive data securely under quantum attacks. To solve that problem foundamently, migration from traditional cryptography to quantum-resistant cryptography is necessary.

Post-Quantum Cryptography (PQC) is regarded as an effective way to resist attacks from quantum computers. Since National Institute of Standards and Technology (NIST) proposed its PQC standardization project in 2016, many candidates have been submitted and the quantum-resistant capabilities of each post-quantum cryptography scheme have been evaluating by researchers. There are several different ways to construct quantum-resistant cryptographic schemes such as lattice-based cryptography [17], hash-based cryptography [15] [3], multivariate-based cryptography [22], code-based cryptography [14], etc. Comparing with traditional digital signature schemes such as RSA or ECDSA, post-quantum digital signature schemes generate much larger key pairs or signatures, and they cost more time for key generating, signing and verification. These features restricted their practicality, especially when applied to the constrained environments. Table 1 shows the candidates of post-quantum digital signature schemes out of round 2 of NIST's PQC project.

Lattice-based cryptography is one of the competitive candidates. Its practicality on constrained environments had

[1] Intelligent Systems Laboratory, SECOM CO.,LTD.
[a)] shu-sho@secom.co.jp
[b)] tadahi-ito@secom.co.jp

been investigated by some researchers. For many lattice-based cryptographic schemes, polynomial multiplication and discrete Gaussian sampling are two main challenges on devices with constrained memory and limited computing power. Albrecht et al. [1] implemented "Kyber" presented in [4] on some smart card platforms by using RSA/ECC co-processor and APIs. Yuan et al. [23] proposed memory-constrained implementation of several lattice-based cryptographic schems on standard Java Card platform by improving Montgomery modular multiplication (MMM) [16] and number theoretic transform (NTT) for polynomial multiplication and modifying several discreta Gaussian sampling algorithms. In the other hand, another factor that is likely to affect the practicality and the efficiency which have not been evaluated much, is about the cryptographic boundary as defined in FIPS 140-2 [12]. To perform trusted cryptographic operations in HSM, the way of applying each component of digital signature schemes should be clearly designed. Most wildly used cryptographic algorithms such as RSA and ECDSA allow separation of hash function and asymmetric operation as default. Sugiyama et al. [21] implemented and evaluated the performance of one of such separable algorithm (TESLA#) on Safenet ProtectServer Network HSM. It indicated that it is possible to applied PQC on HSM. In the other hand, he usage of SHA3 hash function is not separated with asymmetric operation for some lattice-based cryptographic schemes. Because of that, those none-separable implementation of PQC may have limited performance according to varied sizes of input messages. We aim to evaluate the practicality of lattice-based cryptographic schemes which are using SHA3 hash functions on HSM and the detailed will be introduced in section 2 and section 3.

Table 1 Round 2 Candidiates of Post-Quantum Digital Signature Schemes

Type	Signature
Lattice-based	CRYSTALS-DILITHIUM [9] FALCON [11] qTESLA [2]
Hash-based	SPHINCS+ [5]
Multivariate-based	GeMSS [7] LUOV [6] MQDSS [19] Rainbow [8]
Zero-knowledge Proofs	Picnic [24]

1.1 Our Contributions

First, we consider the issue of dealing with cryptographic boundary when applying lattice-based cryptography to HSM, and our works evaluated practicality of lattice-based digital signature schemes by comparing the performances of hash functions operated inside or not inside of the same cryptraphic boundary for HSM.

We also propose the appropriate way to construct cryptographic boundaries for lattice-based cryptographic schemes when applied to HSM.

The rest of this paper is organized as follows. We give a brief mathematical background of lattice and the introduction of HSM and cryptographic boundary. in Section 2. We describe the details of three lattice-based digital signature schemes and analyze the challenge for applying them to HSM in section 3. Evaluation of the results is given in section 4. We then evaluate the cost in section 5. Finally, we conclude the paper in section 6. We believe that our result helps to define cryptographic boundary for PQC, where theoretical proof and clearance of patents should be done.

2. Preliminaries

In this section, we give a brief mathematical description of lattice-based cryptography in section 2.1, we introduce the general way of using digital signature schemes in HSM and point out the challenge of our work in section 2.2, then we introduce the concept of cryptographic boundary which plays very important role at this paper in section 2.3.

Throughout this thesis, bold italic letters denote polynomials (e.g. $\mathbf{f}$ or $\mathbf{F}$), bold lower-case denote vectors (e.g. $\mathbf{v}$), and bold upper-case letters denote matrices (e.g. $\mathbf{A}$). For a set S, we write $s \leftarrow S$ to denote that an element s is chosen uniformly at random from S. if S is a probability distribution, $s \leftarrow S$ denotes that s is chosen according to S. We denote by $\mathbb{R}$ the field of real numbers. Let n and q be a positive integers, we denote by $\mathbb{Z}_q$ the set of integers $0, 1, ..., q-1$, and $\mathbb{R}_q = \mathbb{Z}_q[x]/(x^n+1)$ the quotient polynomial ring such that for any polynomial $\mathbf{f} \in \mathbb{R}_q$, its maximum degree is $n-1$ and coefficients are in $\mathbb{Z}_q$.

2.1 lattice

A lattice is a subgroup of the Euclidean space. Let $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n\} \in \mathbb{R}^{m \times n}$ be a set of linearly independent vectors, the lattice generated by $\mathbf{A}$ is the set $L(\mathbf{A}) = \{\sum_{i=1}^{n} \mathbf{a}_i x_i | x_i \in \mathbb{Z}\}$.

We refer $\mathbf{A}$ as a basis of the lattice $L(\mathbf{A})$, where m and n are the dimension and the rank of the lattice, respectively. In lattice-based cryptography, it is common to consider integer lattices only, i.e. $L(\mathbf{A}) \in \mathbb{Z}^m$. In this paper, we will be concerned with full rank lattices, i.e. $n = m$. Any lattice admits multiple different basises.

Many provably secure lattice-based cryptographic schemes are based on the hardnesss of lattice problems in the worst-case. Besides the most classical problems which are shortest vector problem (SVP) and closest vector problem (CVP), other problems such as learning with error's problem (LWE) [18] or ring learning with error's problem (R-LWE) [13] are also used to construct provably secure cryptographic schemes.

2.2 HSM and Hash Functions

A hardware security module (HSM) is a physical computing device that provides a trusted environment to perform cryptographic operations such as encryption, decryption and authentication, etc. HSMs typically satisfy the FIPS 140-2 [12] and common criteria standard to achieve the high security. Relying on secure mechanisms to create an isolated environment from normal computing environments, HSMs ensure reliable generation, protection, and managment of keys and sensitive data. It is now widely used in some critical infrastructure, for instance, be used as a part of public key infrastructure (PKI) or internet bank infrastructure. In these cases, many HSMs are connected together to preserve high practicality and efficiency.

In most of the current signing systems allow the separation of hash function and asymmetric operation. Figure 1 shows that the message is hashed in message management server and the fixed sizes hash values are sent into HSM. The transmission of the fixed length digest between the message management server and HSM is quite efficient to implement. Then the signature generation is done inside of the HSM. The traditional cryptography such as RSA or ECDSA which is using SHA2 families of hash functions can be combined with HSM in this way.

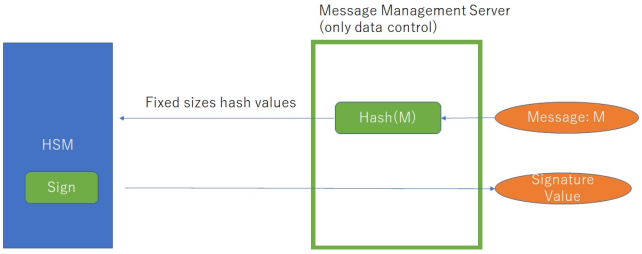

Fig. 1 In general cases, message is hashed in message management server and sent to HSM. The signature generation is finished in HSM.

In the other hand, the alternative of SHA2, which is called SHA3 [10] families of hash functions began to be applied into many PQC cryptographic schemes to improve the security against quantum attacks. In many of those PQC algorithms, random values are generated and hashed with the message together. For some lattice-based cryptographic schemes, the random values are secret or associated with private key. To prevent the leakage of random values, we may introduce HSMs. In that case, whole message (instead of hash value) has to be tranmitted into HSM and therefore the call for hash functions are calculated inside of it. In this case, the random values and message locate in the same level of cryptographic boundary whose introduction is given in section 2.3.

Figure 2 shows the operations, that the flexible size message (which could be very large) is sent to HSM directly. Therefore may require much more resources in HSM. In this paper, we give the experimental results of the differences of the resources cost of hash functions for three lattice-based cryptographic schcmes and propose a way to avoid that increase.

2.3 Cryptographic boundary

Cryptographic boundary was defined in FIPS 140-2 [12]. For a cryptographic module that are used within a cyber system, the cryptographic boundary establishes the physical bounds that contain all the software, hardware, and

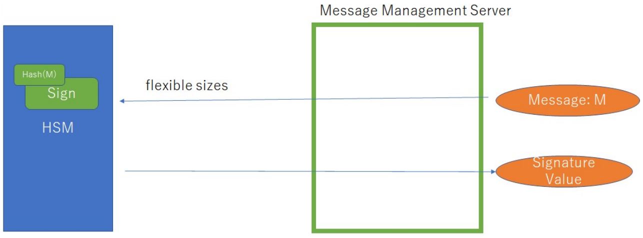

Fig. 2 For some cryptographic schemes in which the message is hashed with secure values, the digest generation and signature generation should be finished in HSM.

firmware of this cryptographic module. It is essential to clearly defined the range of the cryptographic boundary in order to guarantee the security of the cryptographic module.

In this section, we describe method to construct cryptographic boundary for efficient operations also. For example, if we want to implement a system who has similar architecture as Figure 1 or Figure 2 and be executed in a single cryptographic boundary (as illustrated at Figure 3), that access controls need to be prepared for protecting keys in the cryptographic boundary, which tend to be costly. On the other hand, if the cryptographic boundary contains more components of a cryptographic module, operations across cryptographic boundary may increase and therefore a more complex access control mechanisms would be needed, which would increase implementation cost.

It is considered to be more efficient to build more than one cryptographic boundary for a cryptographic module. To be more precise, a system can be implemented and located into several cryptographic boundaries like the way shown in Figure 4. By this implementation, processes related to key objects are stored in the inner cryptographic boundary, and data management of to-be-signed data would be done with access control of outer cryptographic boundary. Benefits of such an implementation include the followings.

- Access control of keys and their metadata should be extremely strict. This implementation can minimize the scope of such strict access control.
- Basically, data flows across the inner cryptographic boundary are fixed size data. Therefore, it is much easier to facilitate data into the cryptographic boundary.
- System migration is accomplished easier. The transition of the whole system can be divided into the migrations of inner boundary and outer boundaries. Although the needs for interoperability, migration can be divided into non-dependent steps, and costs of migration can be reduced.

In section 3 and section 4, we will give the introduction of three lattice-based digital signature schemes and analyse the way of constructing appropriate cryptographic boundaries when applied them to HSM. We discuss the details about migration costs in Section 5.

3. Lattice-based Digital Signature Schemes

In this section, we give a brief introduction of the round 2

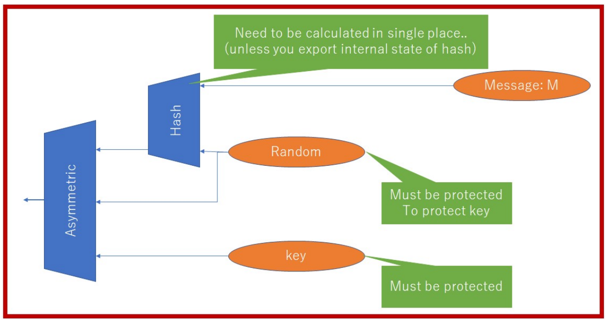
Need to be calculated in single place..
(unless you export internal state of hash)
Message: M
Hash
Random
Asymmetric
Must be protected
To protect key
key
Must be protected

Fig. 3 Message locates in same boundary with signature generation operation for cryptographic schemes in which the message digest is derived by hashing the conjunction of message and some secret values.

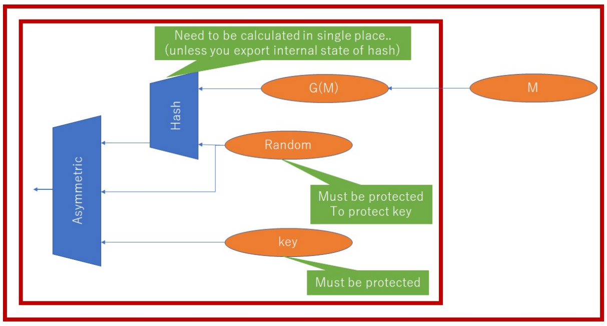
Need to be calculated in single place..
(unless you export internal state of hash)
G(M)
M
Hash
Random
Asymmetric
Must be protected
To protect key
key
Must be protected

Fig. 4 Message locates in different boundary from signature generation operation for those cryptographic schemes that allow the separation of hash function and asymmetric operation.

lattice-based digital signature schemes candidates FALCON, qTESLA and CRYSTALS-DILITHIUM. Then we analyse the practicality of their signature generation operations executing on HSM.

3.1 FALCON

Fouque et al. [11] proposed the fast fourier lattice-based compact signatures (FALCON) which is based on NTRU lattices. Algorithm 1 shows the signature generation of FALCON.

Here are some definitions. Function **HashToPoint**() is based on a SHAKE-256 hash function and refered to algorithm 7 in [11], **ffSampling**() represents the fast fourier sampling algorithm which is refered to algorithm 19 in [11], the compression function **Compress**() is refered to algorithm 21 in [11]. The function **FFT**() is to compute the fast fourier transform representation and **invFFT**() is to compute its' inverse. In step 1 of Algorithm 1, a salt value $\mathbf{r} \in \{0,1\}^{320}$ is derived uniformly at random, and hashed with message m. If a secure execution environment can be provided for the uniformly random generation of $\mathbf{r}$, step 1 and step 2 can be done outside of HSM, then the fixed length digest c is transmitted into HSM to cope with the rest cryptographic functions. Under this circumstances, the cryptographic boundary can be constructed as shown in the right side of Figure 5. Otherwise, step 1 and step 2 execute within the HSM and the cryptographic boundary structure is shown in the left side of Figure 5.

Algorithm 1: Signature Generation of FALCON

```
Input :
        The private key $sk$;
        The message $m$;
        A bound $\beta$;
Output:
        The signature $sig = (\mathbf{r}, \mathbf{s})$;
1  $\mathbf{r} \in \{0,1\}^{320}$ uniformly;
2  $c = \mathbf{HashToPoint}(\mathbf{r}||\mathbf{m})$;
3  $\mathbf{t} = (\mathbf{FFT}(c), \mathbf{FFT}(0)) \cdot \widehat{\mathbf{B}}^{-1}$ ;
4  do
5      $\mathbf{z} = \mathbf{ffSampling}_n(\mathbf{t}, \mathbf{T})$;
6      $\mathbf{s} = (\mathbf{t} - \mathbf{z})\widehat{\mathbf{B}}$
7  while $||s|| > \beta$;
8  $(s_1, s_2) = \mathbf{invFFT}(\mathbf{s})$;
9  $\mathbf{s} = \mathbf{Compress}(s_2)$;
10 return $sig = (\mathbf{r}, \mathbf{s})$
```

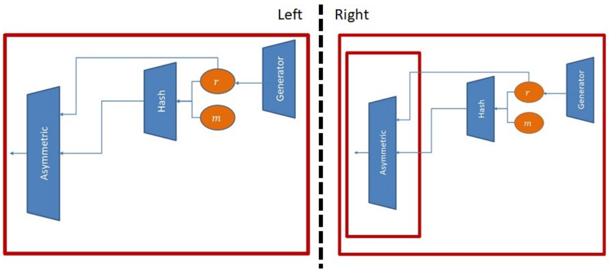

Fig. 5 The structure of cryptographic boundaries for FALCON (Left: the hash operation locates in same cryptographic boundary as asymmetric operations in HSM. Right: the hash operation locates in different cryptographic boundary from asymmetric operations in HSM.)

3.2 qTESLA

Akleylek et al. [2] proposed qTESLA digital signature scheme, which is based on the hardness of the decisional ring learning with errors (R-LWE). Algorithm 2 shows the signature generation for qTESLA.

Here are some definitions. $\mathbf{PRF}_2() : \{0,1\}^k \times \{0,1\}^k \times \{0,1\}^{320} \rightarrow \{0,1\}^k$ is performed as a pseudorandom function. $\mathbf{ySampler}() : \{0,1\}^k \times \mathbb{Z} \rightarrow R_{[B]}$ is refered to algorithm 12 in [2]. $\mathbf{GenA}() : \{0,1\}^k \rightarrow R_q^k$ is refered to algorithm 10 in [2]. In step 3 of Algorithm 2, the function $\mathbf{PRF}_2()$ hashes message m with secret data $\mathbf{seed}_y$ and a random value r, the usage of $\mathbf{G}() : \{0,1\}^* \rightarrow \{0,1\}^{320}$ was first introduced in the Ver. 2.8 (11/08/2019) of [2], which made it possible to transmit the fixed length messgage digest $\mathbf{G}(m)$, instead of m, into HSM. Without the optimization in Ver. 2.8, the original message m is transferred into HSM directly and the structure of cryptographic boundary is shown in the left part of Figure 6. After Ver. 2.8, the cryptographic boundary can be constructed in the way shown in the right part of Figure 6. We will compare the practicalities of these two structures for qTESLA in the next section.

3.3 CRYSTALS-DILITHIUM

Ducas et al. [9] proposed CRYSTALS-DILITHIUM digital signature scheme, which is based on the hardness of the

Algorithm 2: Signature Generation for qTESLA

```
Input :
        The private key
    $sk = (s, e_1, ..., e_k, \mathbf{seed}_a, \mathbf{seed}_y, g)$;
        The message $m$;
Output:
        The signature $sig = (z, c')$;
1  counter = 1 ;
2  $r \in \{0,1\}^k$;
3  rand = $\mathbf{PRF}_2(\mathbf{seed}_y,\ r,\ \mathbf{G}(m))$;
4  $y$ = $\mathbf{ySampler}$(rand, counter) ;
5  $a_1, ..., a_k \leftarrow \mathbf{GenA}(\mathbf{seed}_a)$;
6  for $i = 1, ..., k$ do
7      $v_i = a_i y \bmod^{\pm} q$
8  end
9  $c' = \mathbf{H}(v_1, ..., v_k, \mathbf{G}(m), g)$ ;
10 $c = \{pos_l ist, sign_l ist\} \leftarrow \mathbf{Enc}(c')$ ;
11 $z = y + sc$ ;
12 if $z \notin R_{[B-S]}$ then
13     counter = counter + 1 ;
14     Restart as step 4
15 end
16 for $i = 1, ..., k$ do
17     $w_i = v_i - e_i c \bmod^{\pm} q$ ;
18     if $||[w_i]_L||_\infty \geq 2^{d-1} - E \vee ||w_i||_\infty \geq \lfloor q/2 \rceil - E$ then
19         counter = counter + 1 ;
20         Restart as step 4
21     end
22 end
23 return $(z, c')$
```

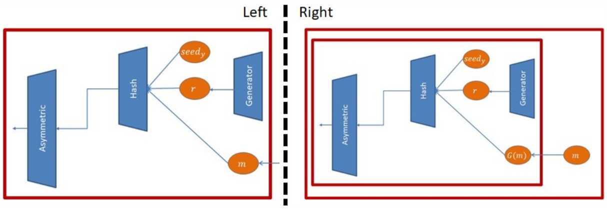

Fig. 6 The structure of cryptographic boundaries for qTESLA (Left: the hash operation locates in same cryptographic boundary as asymmetric operations in HSM for old version. right: the hash operation locates in different cryptographic boundary from asymmetric operations in HSM from Ver. 2.8)

shortest vector problem in lattice.

Here are some definitions. Let $k, l,\ \gamma_1 \in \mathbb{Z}$, q be the moduli, $\mathbf{ExpandA} : \{0,1\}^{256} \rightarrow R_q^{k \times l}$, the hash function $\mathbf{CRH}() : \{0,1\}^* \rightarrow \{0,1\}^{384}$, $\mathbf{ExpandA}()$ maps a seed ρ' and a nonce k to $y = S^l_{\gamma_1 - 1}$. The function $\mathbf{NTT}()$ is to compute the number theoretic transfor representation and $\mathbf{NTT}^{-1}()$ is to compute its inverse. Funtions $\mathbf{HighBits}()$, $\mathbf{Decompose}_q()$ and $\mathbf{MakeHint}_q()$ are refered to figure 3 in [9]. Algorithm 3 shows the signature generation for Dilithium. μ is derived by hashing the conjunction of tr and message M. ρ' is derived by hashing the conjunction of K and μ. Since tr and K are part of the private key, hash function $\mathbf{CRH}()$ need to be calculated inside of HSM and the message M has to be sent to HSM directly. Under this circumstances, the structure of cryptographic boundary is shown in Figure 7.

Algorithm 3: Signature Generation for Dilithium

```
Input :
        The private key $sk = (\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0)$;
        The message $M$;
Output:
        The signature $\sigma = (\mathbf{z}, \mathbf{h}, c)$;
$\mathbf{A} \in R_q^{k \times l} = \mathbf{ExpandA}(\rho)$        ▷$\mathbf{A}$ is generated and
  stored in NTT Representation as $\widehat{\mathbf{A}}$ ;
$\mu \in \{0,1\}^{384} = \mathbf{CRH}(tr||M)$;
$k = 0,\ (\mathbf{z}, \mathbf{h}) =\perp$;
$\rho' \in \{0,1\}^{384} = \mathbf{CRH}(K||\mu)$ (or $\rho' \leftarrow \{0,1\}^{384}$ for
  randomized signing);
while $(\boldsymbol{z}, \boldsymbol{h}) =\perp$ do
    $\mathbf{y} \in S^l_{\gamma_1 - 1} = \mathbf{ExpandMask}(\rho', k)$;
    $\mathbf{w} = \mathbf{Ay}$        $\mathbf{w} = \mathbf{NTT}^{-1}(\widehat{\mathbf{A}} \times \mathbf{NTT}(\mathbf{y}))$ ;
    $\mathbf{w}_1 = \mathbf{HighBits}_q(\mathbf{w}, 2\gamma_2)$;
    $c \in B_{60} = \mathbf{H}(\mu, \mathbf{w}_1)$        ▷ Store $c$ in NTT
      representation as $\widehat{c} = \mathbf{NTT}(c)$ ;
    $\mathbf{z} = \mathbf{y} + c\mathbf{s}_1$        ▷ Compute $c\mathbf{s}_1$ as $\mathbf{NTT}^{-1}$
      $(\widehat{c} \cdot \widehat{\mathbf{s}}_1)$ ;
    $(\mathbf{r}_1, \mathbf{r}_0) = \mathbf{Decompose}_q\ (\mathbf{w}$ - $c\mathbf{s}_2,\ 2\gamma_2)$        ▷
      Compute $c\mathbf{s}_2$ as $\mathbf{NTT}^{-1}$ $(\widehat{c} \cdot \widehat{\mathbf{s}}_2)$;
    if $||\boldsymbol{z}||_\infty \geq \gamma_1 - \beta$ or $||\boldsymbol{r}_0||_\infty \geq \gamma_2 - \beta$ or $\boldsymbol{r}_1 \neq \boldsymbol{w}_1$
      then
        $(\mathbf{z}, \mathbf{h}) =\perp$
    end
    else
        $\mathbf{h} = \mathbf{MakeHint}_q(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$
          ▷ Compute $c\mathbf{t}_0$ as $\mathbf{NTT}^{-1}$ $(\widehat{c} \cdot \widehat{\mathbf{t}}_0)$;
        if $||c\boldsymbol{t}_0||_\infty \geq \gamma_2$ or the# of 1's in $\boldsymbol{h}$ is greater
          than $\omega$ then
            $(\mathbf{z}, \mathbf{h}) =\perp$
        end
    end
    $k = k + 1$ ;
end
return $\sigma = (\mathbf{z}, \mathbf{h}, c)$
```

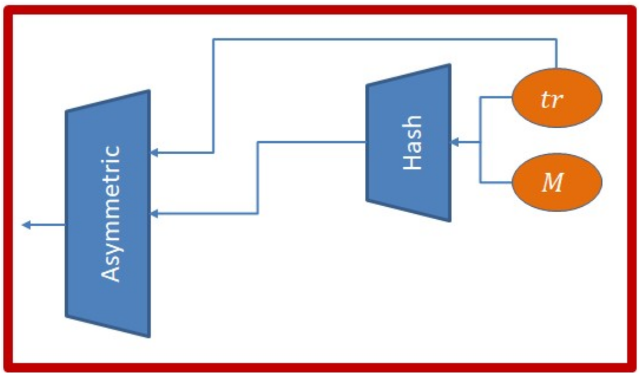

Fig. 7 The structure of cryptographic boundaries for CRYSTALS-DILITHIUM: the hash operation locates in same cryptographic boundary as asymmetric operations in HSM)

4. Evaluation

In this section, we evaluate the practicality of FALCON, qTESLA, DILITHIUM applying to HSM. Hashing operations for three lattice-based digital signature schemes are implemented and be used for evaluating the performances of different cryptographic boundary structures for these digital signature schemes.

4.1 HSM Specification

Throughout the paper, the HSM which we used to do our experiments is the Protect Server External 2 (PSE-2) produced by Gemalto N.V.. ProtectServer Client software version 5.6 is selected as our standard. PSE-2 satisfies FIPs 140-2 level3, and the functional module mechanism makes it possible to implement and execute desired functions on it. Varieties of interfaces defined in PKCS#11, OpenSSL, JCProv, JCA/JCE and MS CAPI/CNG are supported. Moreover, Protect Server Tool Kit (PTK) for C, C# and Java language have been prepared for a more convenient development.

4.2 Experimental Results

We call it boundary type "A" when the message (no matter how long the length is) is sent to HSM directly, and there is only one cryptographic boundary constructed for signature generation, this structure is similar as Figure 3. The boundary type "B" represents that only the fixed length message digest is transfored into HSM, and there is another cryptographic boundary for protecting the generation of message digest from message, the structure is similar as Figure 4. Table 2 shows the time costs for hashing operations executed inside of HSM for FALCON, qTESLA and CRYSTALS-DILITHIUM with different structures of cryptographic boundary. The time is measured as millisecond (ms). The size of message is measured as kilobyes (K) or megabytes (M). For FALCON, when message is hashed in another cryptographic boundary from HSM, the time cost of hash operations inside the HSM is 0. However, for boundary type A, when the message size becomes larger and larger, the time cost raises. If the message size is 10M, the time cost of hash operation is about 11667.19ms($\approx$11.7s). For qTESLA, when boundary type A is applied, the time cost grows with the extension of message size. After the optimization from Ver. 2.8, no matter how long the size of the original message, the fixed size message digest is generated and sent into HSM, therefore, the time cost is almost the same for each case as shown in the second row of the experiment results for qTESLA. For CRYSTALS-DILITHIUM, because that the message digest is derived by hashing the conjunction of message and secret key's components together, all the operations of signature generation locate in the same boundary and therefore the time cost raises with the extension of message size.

5. Migration costs for each cryptographic boundary

RSA (with sha2) and ECDSA (with sha2) signing systems using HSMs typically utilize type B cryptographic boundaries. This section describe migration costs of those signing systems toward lattice-base signatures with cryptographic boundary of type A or type B.

5.1 Migration costs for boundary type A

To migrate to type A, much more components of lattice-based signatures are contain in a single cryptographic boundary, so as a general rule, it is expected that the access control mechanism has to be designed to be more complicated, which is likely to require much more processing resource for access control. These changes should be done after a thorough works of threat analysis, redefining of threat models, and redefining of human operations.

In addition, as described in Section 4, a lot of hash calculations are accomplished inside of the type A cryptogeaphic boundary for key management, much more protected computing resource inside of the boundary may be required to sign large size files.

5.2 Migration costs for boundary type B

To migrate to type B, it is possible to utilize the same kind of cryptographic boundary as used in traditional systems like RSA or ECDSA. In this case, although it is necessary to support the lattice-based cryptographic algorithms with corresponding object ID, the change in architecture of cryptographic boundary between lattice-based and traditional implementation is likely to be limited.

As shown in Figure 8, if it were possible to place both lattice-based and traditional cryptographic modules into the same boundary, moreover, if it were possible to switch the lattice-based-signatures-related inner boundary and traditional-signatures-related inner boundary from inside of their common outer boundary, changes of human operation will also be limited. This approach also require interoperability and standardization of API, but the benefits of success will be great.

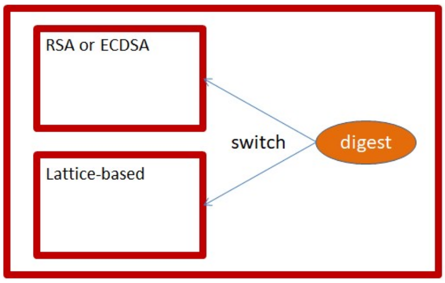

Fig. 8 A mechanism that allows both lattice-based-signatures-related inner boundary and traditional-signatures-related inner boundary to be located inside the same outer boundary

6. Conclusion

In this paper, we discuss the practicality of lattice-based cryptography, which is one of the candidates of NIST's project, when applied to Hardware Security Module (HSM). We describe the features of three lattice-based digital signature schemes selected from round 2 of NIST's project, and point out that the way of using hash function restricts their

Table 2 The time cost for hashing operations executed inside of HSM for three lattice-based digital signature schemes with different structures of cryptographic boundary

Scheme	Boundary Type	Time (millisecond)				
		1k	10k	100k	1M	10M
FALCON	A	33.36	38.08	142.26	1240.59	11667.19
	B	0				
qTESLA	A (before Ver. 2.8)	34.78	44.78	138.05	1196.26	11810.52
	B (from Ver. 2.8)	30.84	38.25	38.63	38.63	31.42
DILITHIUM	A	34.67	45.79	156.19	1351.4	12727.83

practicality by comparing the performances of hash functions processed inside or outside of HSM. We also propose an approach to improve the practicality of PQC when applied to HSM. We believe that our result helps to define cryptographic boundary for PQC, where theoretical proof and clearance of patents should be done.

One of the future work is that, we only compared bottleneck when apply three lattice-based digital signature schemes on HSM, a total comparison of time cost for signature generation can be done to evaluate the change when optimize the usage of hash functions.

References

[1] Martin R. Albrecht, Christian Hanser, Andrea Hoeller, Thomas=Pöppelmann, Fernando Virdia, and Andreas Wallner. "Implementing RLWE-based schemes using an RSA co-processor." In Cryptology ePrint Archive, 2018/425, 2018.

[2] Nina Bindel, Sedat Akeylek, Erdem Alkim, Paulo SLM Barreto, Johannes Buchmann, Edward Eaton, Gus Gutoski, Julaine Kramer, Patrick Longa, Harun Polat, Jefferson E. Richardini, and Gustavo Zanon. qtesla. submission to the nist's post-quantum cryptography standardization process.(2018), 2018.

[3] Johannes Buchmann, Erik Dahmen, and Andreas Hülsing. "XMSS - a practical forward secure signature scheme based on minimal security assumptions." In Bo-Yin Yang, editor, *Post-Quantum Cryptography*, volume 7071 of *Lecture Notes in Computer Science*, pp. 117-129. Springer Berlin / Heidelberg, 2011.

[4] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, and Damien Stehlé. "CRYSTALS - Kyber: a CCA-secure module-lattice-based KEM." In IACR Cryptology ePrint Archive, Report 2017/634, 2017.

[5] Daniel J. Bernstein, Andreas Hülsing, Stefen Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. "The SPHINCS+ Signature Framework." submission to the nist's post-quantum cryptography standardization process, 2019. `https://sphincs.org/data/sphincs+-paper.pdf`.

[6] Ward Beullens, Bart Preneel, Alan Szepieniec, and Frederik Vercauteren. LUOV, Csrc.nist.gov, Jun 2019, [online] Available: `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions`.

[7] A. Casanova, J.-C. Faugere, G. Macario-Rat, J. Patarin, L. Perret, and J. Ryckeghem. "GeMSS: A great multivariate short signature." Jun 2019, [online] Available: `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions`.

[8] Jintai Ding, Ming-Shing Chen, Albrecht Petzoldt, Dieter Schmidt, and Bo-Yin Yang. Rainbow specifications. NIST PQC Round 2 Submission (2019).

[9] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "CRYSTALS-Dilithium: A lattice-based digital signature scheme." *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(1):238-268, 2018. `https://tches.iacr.org/index.php/TCHES/article/view/839`.

[10] Morris J. Dworkin. "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions." `https://www.nist.gov/publications/sha-3-standard-permutation-based-hash-and-extendable-output-functions?pub_id=919061`, 2015.

[11] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. "Falcon: Fast-fourier lattice-based compact signatures over ntru." submission to the nist's post-quantum cryptography standardization process.(2018), 2018.

[12] FIPS 140-2. "Security Requirements for Cryptographic Modules." `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf`.

[13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors Over Rings." In IACR Cryptology ePrint Archive, Report 2012/230, 2012.

[14] Robert J. McEliece. "A public key cryptosystem based on algebraic coding theory." *DSN progress report*, 42-44:114-116, 1978.

[15] Ralph Merkle. "A certified digital signature." In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO?89 Proceedings, volume 435 of Lecture Notes in Computer Science*, pp. 218-238. Springer Berlin / Heidelberg, 1990.

[16] Peter L. Montgomery. "Modular multiplication without trial division." In *Mathematics of Computation*, Vol. 44, No. 170, pp. 519-521, 1985.

[17] Daniele Micciancio and Oded Regev. "Lattice-based cryptography." In *Post-Quantum Cryptography*, pp. 147-191. Springer, 2008

[18] Oded Regev, "On lattices, learning with errors, random linear codes, and cryptography." In Proceedings of the thirty-seventh annual ACM symposium on Theory of computing (Baltimore, MD, USA: ACM, 2005), 84-93, `http://portal.acm.org/citation.cfm?id=1060590.1060603`.

[19] Simona Samardjiska, Ming-Shing Chen, Andreas Hulsing, Joost Rijneveld, and Peter Schwabe. MQDSS specifications. NIST PQC Round 2 Submission (2019).

[20] Peter Williston Shor. "Algorithms for quantum computation: discrete logarithms and factoring." In *Proceedings of the 35th Annual Symposium on Fundamentals of Computer Science (FOCS)*, pp. 124-134, 1994.

[21] Shotaro Sugiyama, Tadahiko Ito, and Kohei Isobe. "Implementation and Evaluation of Post Quantum Cryptography on Hardware Security Module." In *2019 Symposium on Cryptography and Information Security*, Shiga, Japan. Jan. 22-25, 2019.

[22] Matsumoto Tsutomu and Imai Hideki. "Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and MessageEncryption." *Lecture Notes in Computer Science.* Berlin / Heidelberg. Springer, 1988.

[23] Ye Yuan, Kazuhide Fukushima, Junting Xiao, Shinsaku Kiyomoto, and Tsuyoshi Takagi. "Memory-Constrained Implementation of Lattice-based Encryption Scheme on the Standard Java Card Platform." In IACR Cryptology ePrint Archive, Report 2018/1238, 2018.

[24] Greg Zaverucha, Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Jonathan Katz, Xiao Wang, Vladmir Kolesnikov, and Daniel Kales. Picnic. submission to the nist's post-quantum cryptography standardization process, 2018. `https://microsoft.github.io/Picnic/`.