Challenges of Post-Quantum Digital Signing in Real-world Applications: A Survey*

Teik Guan Tan, Pawel Szalachowski, and Jianying Zhou

Singapore University of Technology and Design
teikguan_tan@mymail.sutd.edu.sg
pjszal@gmail.com
jianying_zhou@sutd.edu.sg

Abstract. Public key cryptography is threatened by the advent of quantum computers. Using Shor's algorithm on a large-enough quantum computer, an attacker can cryptanalyze any RSA/ECC public key, and generate fake digital signatures in seconds. If this vulnerability is left unaddressed, digital communications and electronic transactions can potentially be without the assurance of authenticity and non-repudiation. In this paper, we study the use of digital signatures in 14 real-world applications across the financial, critical infrastructure, Internet, and enterprise sectors. Besides understanding the digital signing usage, we compare the applications' signing requirements against all 6 NIST's post-quantum cryptography Standardization round 3 candidate algorithms. This is done through a proposed framework where we map out the suitability of each algorithm against the applications' requirements in a feasibility matrix. Using the matrix, we identify improvements needed for all 14 applications to have a feasible post-quantum secure replacement digital signing algorithm.

Keywords: Digital Signing · Post-Quantum Cryptography · Public Key Cryptography · NIST Standardization.

1 Introduction

The use of asymmetric key cryptography to create digital signatures as a means of authentication and non-repudiation is pervasive. On any given day, millions of web servers use digital signing as part of the Transport Layer Security (TLS) [52, 82, 83] to allow users to verify the identity of the server. Millions of merchant payment terminals verify the digital signatures from tens of billions of Europay-Mastercard-VISA (EMV) [28] payment cards to ascertain that the cards are not cloned. And hardware and software vendors rely on digital signatures to protect the integrity of firmware, libraries, operating system software, and applications running on the billions of servers, laptops, mobile phones, and other devices.

* Accepted for publication in *International Journal of Information Security, Springer.*

Digital signatures are also critical for the legal validity of electronic documents. In the European Union, the Electronic Identification, Authentication and trust Services regulation N° 910/2014 recognizes the use of digital signing to ensure the integrity of electronic documents and equates electronic signatures to the legal equivalent of hand-written signatures. Similarly, the Electronic Signatures in Global and National Commerce Act 2000 in the United States allows for the recognition of electronic documents as legally acceptable, provided that the electronic document is "retained and accurately reproduced for later reference", a property that can be achieved using digital signatures.

Most digital signatures are implemented using Rivest-Shamir-Adleman (RSA) [85] or Digital Signature Algorithm (DSA)[1] [32]. Yet, all RSA and DSA implementations face a potentially catastrophic vulnerability in the face of quantum computers. Using Shor's algorithm [88, 79], an attacker in possession of a large-enough quantum computer can cryptanalyze the RSA or ECC public key and obtain the corresponding RSA or ECC private key in polynomial-time. National Institute of Science and Technology (NIST) stated in its 2016 report [20] that by 2030 with a budget of 1 billion dollars, an RSA-2048 bit key can likely be broken by a quantum computer in a matter of hours.

NIST is spearheading the efforts through a post-quantum cryptography (PQC) Standardization competition [72] to solicit and evaluate possible post-quantum secure digital signature algorithms for use beyond 2030. From NIST's updates [68, 69], we understand that the effort is likely more of a standardization plan of multiple algorithms instead of picking a single winner, since each algorithm has its strengths and weaknesses. For example, hash-based cryptography is quantum-resistant [40, 12], but the limited number of signatures per private key using hash-based signatures [6] and the relatively large overheads in signing make it only suitable in some implementations (such as code-signing) that require a long lifetime, are difficult to transition, and need to be urgently deployed [21]. In looking at applicability, post-quantum TLS authentication has been comprehensively studied by Sikeridis et. al. [89] and the team at Open Quantum Safe (https://openquantumsafe.org/). However, no study investigates other applications of digital signatures. In this paper, we fill this gap, by asking and answering the following questions: but how about other digital signature use-cases? Is it possible to select a suite of signature schemes that can fulfill all the digital signing requirements presently used by applications in the field? Or are we faced with a technology gap where some applications will be left without a viable digital signing solution in the face of quantum computers?

1.1 Methodology

Our approach to investigate how digital signatures can continue to be used in the post-quantum era is a three-step process to i) identify applications that

[1] Elliptic curve cryptography (ECC) [56] is commonly used as the underlying cryptosystem for DSA. For purposes of this paper, we will use Elliptic-Curve Digital Signature Algorithm (ECDSA) to represent all DSA signature implementations using ECC.

use digital signatures; ii) collate the digital signing operating requirements and constraints of these applications; and iii) cross-reference the requirements against NIST's PQC Standardization round 3 algorithms. To ensure comprehensiveness and completeness of our survey, we adopt the following methodology in our selection:

- *Applications.* We want to capture a representative range of digital signature use-cases that is relevant to businesses and end-users. Since the financial, government and telecommunication domains have the largest security expenditure, we set out to search these domains (using Google Search) to identify digital signing applications that have at least one billion in transaction volume or value as the criterion. The assumption is that these applications will continue to use digital signatures in the future, and will be materially impacted if their digital signature implementations are not quantum secure.
- *Digital signature schemes.* After two intensive rounds of evaluation, NIST has shortlisted 6 candidate signature algorithms (out of an initial 19) consisting of 3 finalists and 3 alternates for the final round 3 evaluation [69]. While we recognize that there are other techniques in achieving post-quantum digital signing [63, 92, 21], we choose to only use all 6 candidate algorithms from the NIST PQC standardization round 3 for our evaluation matrix as all 6 have been thoroughly studied and at least one of them will be made into a standard and be widely adopted.
- *Related works.* We do a systematic literature review on Google Scholar of research related to application testing and implementations with post-quantum digital signatures, and exclude algorithms that are not part of NIST's round 3 candidates.

1.2 Contribution and Organization

Our contributions are:

- We present a survey of 14 real-world application use-cases across 4 broad categories to understand how digital signing is deployed and used.
- We propose a framework on different digital signature operating constraints and measure the NIST PQC Standardization candidate algorithms' suitability in meeting the requirements.
- We apply the framework to build a feasibility matrix of application versus signature scheme and show that the NIST PQC Standardization may not yield a suitable post-quantum secure signature algorithm for chip-card-based applications, unless improvements are made to increase the speed of signing and reduce the size of the signatures.

The paper is organized to survey 14 real-world applications that use digital signatures in Section 2. We model the digital signing operating constraints into a proposed requirements framework in Section 3 and proceed to apply the framework in Section 4 to build a feasibility matrix to identify viable algorithms for each application. We conclude in Section 5. A background of digital signing can be found Appendix A, and a brief description of the NIST PQC algorithms under evaluation is found in Appendix B.

2 Digital Signing in Practice

In this section, we examine the use of digital signatures by 14 real-world applications across 4 broad categories (see Figure 1). The categories chosen are financial (for the economy), critical infrastructure (for the government, people, and devices), Internet (for business-to-business, business-to-consumer, peer-to-peer, and Internet-of-things interactions), and enterprise (for businesses). Whilst this is not a complete coverage of all asymmetric cryptography use-cases, it includes all of the digital signing use-cases found in the European Telecommunication Standards Institute report [30] and an extensive use-case search we conducted on Google's search engine. Only military applications are omitted from this study since such requirements are typically classified and scant on public details. For each application, we include a paragraph to discuss the impact on the application if the digital signature scheme is compromised due to quantum computers.

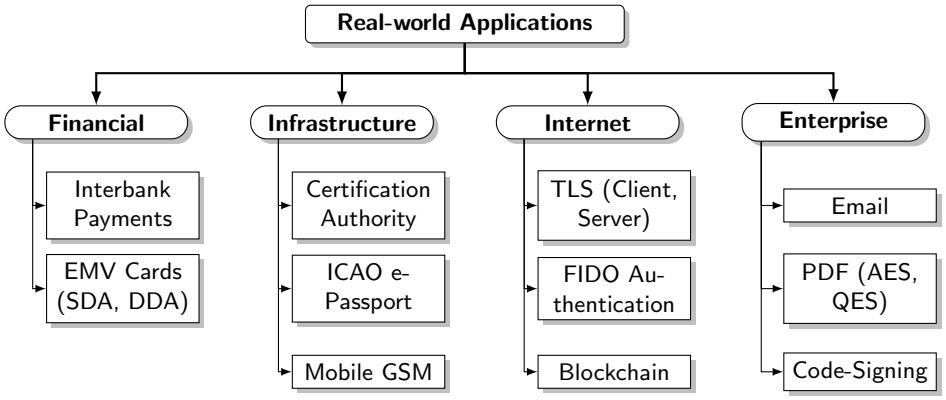

Fig. 1. 14 applications that use digital signatures.

2.1 Financial

Banking and financial institutions deploy strong security to build trust with their customers and maintain their reputation as trustworthy organizations.

Inter-bank Payment Systems On a daily basis, banks transfer hundreds of millions or billions dollars' worth of money electronically amongst each other on the SWIFT (Society for Worldwide Interbank Funds Transfer) inter-bank payment network. Banks have specialized systems and terminals to connect into SWIFT to send and receive payment instructions where mutual authentication and encryption are provided for. Non-repudiation of messages originating from the banks' customers is supported in SWIFT through their 3SKey offering [90] where businesses and end-users will own a 3SKey USB token to participate in

a public key infrastructure (PKI) operated by SWIFT. The signing algorithm used in 3SKey is RSA-2048 [90].

Impact: 3SKey relies on digital signing for data integrity and non-repudiation of the message. A compromise of the digital signature scheme would mean an attacker could issue fake payment instructions and steal money from the bank.

EMV Cards EMV [28] is the chip-based standard that is used for consumer payments. Consumers are provided with an EMV chip-card that is used to pay for purchases at physical merchant stores, while merchants have specialized terminals that can read the EMV chip-cards and obtain the necessary authentication and transaction authorization from the issuing banks via the payment networks. While the majority of the cryptographic algorithms used in EMV are symmetric-key based, EMV relies on asymmetric key digital signatures in the static data authentication (SDA) and dynamic data authentication (DDA) protocols to reduce payment card fraud. EMV cards expire every 3-5 years where they are replaced with new keys.

SDA is the entry-level EMV mechanism to deter fraudsters from illegally generating fake card data to fool merchants. It achieves this by requiring the issuing bank to digitally sign the personalization information embedded in the card. When the consumer presents the card to the merchant, the terminal can verify the bank's digital signature before processing the transaction. DDA is the enhanced EMV mechanism to further prevent offline card fraud. It includes all the features of SDA, with added functionality for the EMV card to digitally sign a challenge from the merchant terminal to defeat offline card cloning fraud between the merchant terminal and EMV card. The EMV signing algorithm is RSA of up to 1984 bits, due to the 255-byte maximum size of an application protocol unit (APDU) defined in ISO-7816-4 [51], and the hashing used is SHA-1 [28].

Impact: Both EMV-SDA and EMV-DDA rely on digital signing for authentication of the card by the merchant terminal. A compromise of the digital signature scheme would mean an attacker could arbitrarily create, rather than having to clone, fake EMV cards that can authorize transactions in offline mode. In addition, it also renders EMV-DDA cards equally insecure as EMV-SDA cards in offline mode. Transaction authorization in online mode is less affected.

2.2 Infrastructure

The security of public and governmental infrastructure affects the community that relies on such systems for their day-to-day activities.

Certification Authority Certification Authorities (CA) play a key role in providing the root-of-trust or trusted third-party service between different entities. A CA issues digitally signed certificates that attest the entity's identity and other attributes to an associated public key. The CA is also responsible for digitally signing revocation data to inform the network in the event that specific certificates become no longer trustworthy or are compromised. Parties performing the verification of the certificate require an additional step of checking the cer-

tificate against the CA's signed certificate revocation list or by using the online certificate status protocol to get a signed response on the status of the certificate.

Impact: The CA relies on digital signing to ensure the integrity of the identity and public key in the certificate, and this trust is inferred for any application relying on the certificates within the chain for authentication or non-repudiation. The DigiNotar CA hacking incident [93] in 2011, where Google's, Yahoo's, and other web servers were compromised, provides us a glimpse into the potential implications when a CA key is stolen. A compromise of the digital signature scheme would be worse as attackers can create fake certificates and revocations that violate the trust and cause complete failure to any authentication or non-repudiation processes, without any means of remediation through key re-issuance.

ICAO e-Passport International Civil Aviation Organization (ICAO) maintains the ICAO 9303 [48] standard for machine-readable travel documents or e-Passports which is the officially recognized standard used today for identifying persons in all cross-border travel. The e-Passport contains a data page that includes both visual information as well as electronic information (including personal particulars, facial & fingerprint biometric, visa, authentication keys) embedded within the contactless chip-card about the person identified by the document. The electronic information in the e-Passport is signed by a document signer key which is certified by the country signing certification authority (CSCA). CSCAs are root CAs whose self-signed certificates are uploaded into the ICAO public key directory for dissemination to other countries. Document signer keys are short-term usage keys to sign the electronic information but need to remain secure for the entire lifetime (5-10 years) of the e-Passport.

Impact: ICAO 9303 relies on digital signing to assert the integrity of the traveler's identity contained within the e-Passport. A compromise of the digital signature scheme would mean fake passports and identities could be easily created, and this will certainly be a nightmare for travel and border controls.

Mobile GSM Global System for Mobile communications Association (GSMA) is a trade grouping of technology vendors, network operators, and service providers that define, implement and certify standards used for the modern-day mobile communication using handsets. Much of the security related to mobile communications use industry best practices including AES-128 bit encryption, SHA-256 hash and key derivation, and common-criteria assurance level for subscriber identity module (SIM) and embedded SIM (eSIM) cards. Within the GSM eSIM [41] standard where handsets can be dynamically provisioned with different profiles, digital signatures are used to provide mutual authentication between the mobile network operator (MNO) and the Universal Integrated Circuit Card (eUICC), a chip-card equivalent that is embedded within the subscriber's handset.

Impact: GSM eSIM relies on digital signing to ensure both MNO and eUICC are authenticated. A compromise of the digital signature scheme would allow an adversary to masquerade as genuine subscribers to illegally obtain their profiles and carry out account-takeover attacks.

2.3 Internet

The Internet has seen the highest growth in the types of applications, volume of transactions, and number of users in the past 2 decades.

Transport Layer Security (TLS) Transport Layer Security (TLS) [83], preceded by Secure-Sockets Layer (SSL), is the most widespread application-agnostic cryptographic protocol for implementing a secure channel in the client-server model. Its most prominent use is to secure the Hyper-text Transfer Protocol (HTTP) protocol (HTTP over SSL/TLS is referred to as HTTPS). TLS consists of many subprotocols that realize different functionalities. One of them is the handshake (sub)protocol which is responsible for establishing a shared symmetric key between the communicating parties. The high-level TLS v1.3 handshake protocol which performs server authentication, mutual key exchange, and optionally client authentication happens as follows:

1. The client issues a "Client-Hello" message with cipher preferences and one-time key exchange information.
2. The server chooses the appropriate cipher-suite and computes the session key. It responds with a "Server-Hello" and includes key exchange information and a signed response to the client challenge. The server may also optionally request for the client certificate for mutual authentication over TLS.
3. The client, upon receiving the server certificate and signature, will verify the signature and complete the session key exchange.
4. If mutual authentication is requested, the client responds to the server with the client signature and certificate to prove its identity, and this is verified by the server.

Post-quantum TLS extensions have been extensively studied by Sikeridis et al. [89]. The paper worked on selected NIST's PQC Standardization round 2 algorithms by measuring both the speed of the cryptographic computations as well as communication overheads, and compared the performance to RSA-3072 bits at scale. The results show that Dilithium and Falcon, both lattice-based algorithms, perform favorably compared to the other algorithms. We use the performance results from this paper in our framework in Section 3.

Impact: Communicating parties rely on the digital signing function in TLS to authenticate the other party. A compromise of the digital signature scheme would mean adversaries can easily carry out man-in-the-middle attacks to eavesdrop or modify the interaction between the communicating parties.

Blockchain The rise of blockchain started with Nakamoto's Bitcoin white paper [71] in 2008. Since then, we have witnessed a meteoric rise in both resources used and variations of blockchains, mainly in the area of crypto-currencies. The Bitcoin architecture is a distributed system of peer nodes that store and maintain the chain of transaction blocks since the genesis block that was generated by Nakamoto in 2008. Each Bitcoin block contains a collection of transactions, each digitally signed by participating nodes and collated into a Merkle tree structure. To add a block into the Bitcoin blockchain, a miner node will verify the validity of each transaction, including the node's available balance for spending and the

digital signature used to sign the transaction, before completing a proof-of-work consensus protocol to submit the block. The algorithm used for signing transactions is ECDSA with the secp256k1 elliptic curve [80]. Besides using digital signatures to protect transactions, there are variant blockchain implementations that rely on digital signatures in the consensus protocol such as proof-of-stake consensus found in Ouroboros [54] and Dfinity [43]. In this case, chain-growth relies on the block proposer to digitally sign the proposed transaction block to be verified by other nodes in the blockchain.

Impact: Bitcoin relies on digital signing to achieve transaction integrity and non-repudiation for the transfer of Bitcoins between nodes. A compromise of the digital signature scheme will allow adversaries to arbitrarily create valid transactions to steal Bitcoin from other nodes. The use of extensions such as hierarchical-deterministic wallets for one-time use private keys is not a viable post-quantum strategy as Chalkias [17] has shown numerous situations where the public key may be exposed prior to completing a valid transaction despite best efforts.

FIDO Authentication To reduce the reliance on passwords as the primary means of user authentication, the Fast Identity Online (FIDO) alliance proposed the Universal Authentication Framework (UAF) standard [62] that replaces the use of a secret password with biometric authentication. The FIDO client functions as an authenticator to verify the identity of the user, typically using facial or fingerprint biometrics. It then uses a stored private signing key (using ECDSA, RSA or SM2) to sign a challenge as part of the authentication process from the FIDO server.

Impact: The FIDO Server relies on digital signing to remotely authenticate the FIDO client. A compromise of the digital signature scheme would mean adversaries can pose as valid FIDO clients and spoof the FIDO authentication process to carry out account-takeover attacks.

2.4 Enterprise

Digital signatures are also used to provide authenticity and integrity of users and data for applications run within organizations and between interacting organizations.

Email The Secure/Multipurpose Internet Mail Extensions (S/MIME) standard, now in version 4 [87], defines an interoperable standard for how messages can be securely exchanged between two peers. It is commonly used in email applications where the sender has the option to digitally sign the message (for authentication, integrity, and non-repudiation) and/or encrypt the message (for confidentiality and integrity). Both Microsoft and Google, the two leading enterprise email providers, have included S/MIME support in their respective Outlook and G-Suite offerings, giving secure email access to millions of business users.

Impact: S/MIME relies on digital signing to provide message integrity and non-repudiation. A compromise of the digital signature scheme would mean that

communicating parties need to rely on encryption to retain the integrity of the email message, but not prove the non-repudiatable origin of the message.

PDF Documents The push towards secure electronic documents saves costs for the organization, improves productivity for the business processes, and reduces wastage for the environment. ISO 32000 [50] describes how the digital signature (either RSA up to 4096 bits, DSA up to 4096 bits) is to be PKCS#1 signed and PKCS#7 packed and embedded into a portable document format (PDF) document. Adobe has announced that their v11.x products [1] support ECDSA (NIST curves) but no PQC algorithms are planned in the pipeline.

The PDF Advanced Electronic Signature (PAdES) standard [29] describes different levels of assurance for signed PDF documents. PDF documents signed using advanced electronic signatures (PDF-AES)[2] provide the assurance that the document has not been modified since it was signed, but not signer non-repudiation. Typical PDF-AES implementations use a document server in the backend to digitally sign the document with a common signing key protected by a HSM after authenticating the signer. For a higher level of assurance, qualified electronic signatures (PDF-QES) require the signer to be in possession of the private key used for signing and provides for non-repudiation in addition to document integrity. In PDF-QES implementations, signers will be issued with a USB token to store the private key that is used for signing.

Impact: PDF signing relies on digital signing to provide document integrity (for PDF-AES) and signer non-repudiation (for PDF-QES). A compromise of the digital signature scheme will allow adversaries to modify document contents or forge back-dated documents that claim to originate from certain persons. Since documents may contain agreements that have long-term validity periods, the integrity of present-day signed electronic documents may be called into question during the post-quantum era.

Code-signing To ensure that only authorized code is executed, all major operating systems support code-signing as a means to allow manufacturers and publishers to protect the integrity of the applications, system modules, libraries, and firmware when installed or updated on the end-users' devices. Code-signing can also be used in remote attestation where the system proves its trustworthiness to a remote server, for instance by using a Trusted Execution Environment such as Intel SGX to sign the hashes of all applications that are run locally. Operating systems also rely on code-signing to ensure a secure boot process where only signed system modules are loaded and run. Beyond laptops and mobile phones, signature verification on IoT and other terminal platforms need to be taken into consideration as more manufacturers are relying on signed firmware as part of their boot-up sequence.

Impact: Code-signing relies on digital signing to ensure the integrity of the software and that it originates from the manufacturer. A compromise in the digital signature scheme will allow adversaries to inject malicious code into the

[2] This refers to advanced electronic signatures and is not to be confused with the symmetric key encryption standard, advanced encryption standard

software, and distribute the compromised software to the devices to cause greater harm. The Stuxnet attack [58] is one clear example of how attackers were able to exploit a code-signing vulnerability to propagate malicious software that eventually affected a country's nuclear program.

3 Digital Signing Requirements Framework

NIST has listed 3 broad criteria in their call for proposals [72], namely security, cost, and simplicity, of their evaluation for a suitable post-quantum digital signing candidate. Within the cost criterion, specific parameters include the size of keys, computational efficiency of operations, and possibility of computational errors. In round 2 [68, 69], the algorithms were further evaluated based on security (against classical and quantum attacks), performance, and other properties such as resistance against side-channel attacks, flexibility, etc.

Of the 82 post-quantum signing and key-exchange submissions that were received by NIST, 69 were shortlisted for round 1 and a further round 2 shortlist narrowed the field to 26 candidates [4]. The list was further narrowed down to 15 candidates in round 3 [69]. Of the 15, there are a total of 6 candidate signature schemes of which 3 (Dilithium, Falcon, and Rainbow) have been identified as finalists while the other 3 (GeMSS, Picnic, and SPHINCS$^+$) are alternates. Appendix B covers the description of the algorithms.

The intention in this section is not to repeat the evaluation, but to establish a comparable feasibility basis between what a signature scheme is designed for versus what the real-world application needs, on the assumption that the signature scheme remains secure. For example, if a signature scheme determines that key sizes need to be higher than 10,000 bits in size to achieve 128-bit security, but the application can only support key sizes of 2,048 bits due to restrictions within the platform, then the signature scheme is not suitable for the application, regardless of the security or operational efficiency. We first identify the operating constraints which form the requirements criteria for our evaluation and correlate them to the platforms that are used, before matching them against each of the 14 applications.

3.1 Operating Constraints

In our study of applications in Section 2, we observe that the differences in requirements between the digital signature use-cases and algorithms can be distinguished based on execution time and size of cryptographic material.

Execution Time . This constraint determines the execution time of the $KeyGen$, $Sign$, and Ver functions. The higher the throughput requirement from applications, the smaller the execution time per function must be, but the ability to meet the execution time also depends on which device the function is run on. From our study, we observe 3 broad ranges of execution time which we measure in milliseconds (ms):

- *Execution time $\leq$ 1 ms.* We expect to see such throughput in TLS and other server-side authentication applications where the platform is expected

to authenticate thousands of incoming client connections at peak loads. This higher bound is also similar in AWS CloudHSM's highest performance throughput of 1,100 RSA 2048 sign/verify operations per second. Applications whose signing functions need to complete in less than 1 ms impose the biggest constraint.

- *Execution time ≤ 100 ms.* This range is likely to include client-facing use-cases that perform multiple signing functions per second. Each signing function takes up to 100 milliseconds (ms) to complete and is perceived as fast [67] to the end-user. We have to consider that such speeds may be easily accomplished on server platforms but become more challenging on smaller chip-card platforms whose clock speed is 1,000 times slower.
- *Execution time ≤ 1,000 ms.* Applications that call the signing functions only once or very infrequently impose the least constraint. For usability reasons, we expect the function to complete within 1 second.

Size of Cryptographic Material . The maximum allowable sizes of the private key K_s, public key K_p, and signature σ affect the suitability of the signature scheme for use within the application. They are impacted by the application's transmission protocol, communication bandwidth as well as the platform's storage capacity to store the signature and protect the keys. Applications that can support large key and signature sizes have the least constraints while signature schemes with large keys or signature sizes are the most inflexible. Here, we define the following 4 size ranges:

- *Size ≤ 2 Kbits.* This size represents the size of a single APDU packet for chip-cards as defined under ISO 7816-4 [51]. Any data above 2 Kbits would require multiple APDU packets to be exchanged and significantly increases the chances of a transaction failure. Applications that require key and signature sizes to be smaller than 2 Kbits impose the highest constraints.
- *Size ≤ 16 Kbits.* This size represents the largest key and signature size provided for under NIST's Recommendation for Key Management [10]. This is an RSA key with modulus size 15,360 bits and has a security equivalence of 256 bits, excluding any post-quantum considerations. We expect that this value serves as the sizing basis for existing application designers who want to ensure that their applications support the maximum security mechanisms available.
- *Size ≤ 512 Kbits.* This size represents the maximum extended length supported by ISO 7816 interindustry command [51]. Any application using this protocol to communicate with the cryptographic device will not be able to package more than 65,535 bytes of data to and from the device, even if multiple APDU messages are used. This is also the largest data buffer size supported by AWS CloudHSM. Applications that can support such key or signature sizes are considered to impose a small constraint.
- *Size > 512 Kbits.* Applications that can support much larger key and signature sizes beyond 512 Kbits provide virtually no constraint in choosing the appropriate signature scheme, while signature schemes whose keys or signa-

ture sizes exceed 512 Kbits will impose significant storage and transmission overheads when deployed.

3.2 Platform Considerations

The platform on which the signing functions happen also needs to be accounted for. During a digital signing life-cycle, an application uses 3 distinct platforms (P1 → P2 → P3) to carry out the key generation, signing, and verification process:

P1: *To perform all private key K_s operations.* The function $KeyGen$ creates a new private key K_s, and requires the platform to protect the private key in storage. This same platform is where the $Sign$ function is called to generate the signature σ. Each application's P1 platform affects the size of K_s, σ as well as the execution time of $KeyGen$ and $Sign$.

P2: *To output the signature σ.* Applications that use signatures for authentication and non-repudiation such as TLS and 3SKey require a different signature for every interaction. In this case, the platform used for P1 = P2. On the other hand, applications that rely on the integrity property of digital signatures such as EMV-SDA, PDF-AES, and code-signing require the data to be signed only once on platform P1 and have the signature to be stored within the platform P2 to be reproduced on request. In the latter case, the platform used in P2 may not be the same as P1. Each application's P2 platform imposes a constraint on the size of σ.

P3: *To verify the signature σ.* The Ver function runs on the verifying party's platform which is typically different from the proving party in client-server communications. It may also rely on external directories or repositories to retrieve the public key certificate (in the case of CA) or store the signature (in the case of Bitcoin). Each application's P3 platform affects the size of K_p and σ as well as the execution time of Ver.

Table 1. Comparison of selected processors [27]

Processor	Platform	CoreMark	Memory
NXP Semiconductors LPC552x	Chip-cards	330	256-512 KB
STMicroelectronics STM32L552	Chip-cards	443	256-512 KB
Qualcomm Snapdragon 820	Tablets	6,833	Up to 8 GB
Intel Atom E3827	HSMs	10,820	Up to 8 GB
Broadcom BCM2837	Terminals	15,363	Up to 1 GB
Samsung Exynos 5422 / Cortex-A15	Mobile Phones	30,153	Up to 2 GB
Intel Core i5-8250u	Laptops	123,666	Up to 32 GB
AMD Ryzen 1700X	Server	309,792	Up to 64 GB

Column 3 of Table 1 shows the CoreMark [27] performance benchmark comparison of selected processors used in the various platforms, ranging from:

– *EMV chip-cards, USB Tokens, and eUICC.* Due to size (e.g. ISO 7816 requires chip-cards to be able to fit in a wallet) and other physical restrictions,

these platforms are relatively low in both computational power and clock speeds. The CoreMark performance comparison shows that chip-cards perform at roughly 0.1% to 0.2% of the overall compute throughput compared to laptops and servers, and this limits the range of signature schemes that can be supported on these platforms. On the other hand, chip-cards provide a high level of portability and protection for users to transport and use their private keys securely. USB Tokens and eUICC typically share the same internal cryptographic security module as chip-cards, but the input/output communication is not as restricted. As a result, USB tokens and eUICC can handle slightly larger signature (i.e. $\leq$ 16Kbits) sizes. Applications that require to perform signing functions on these platforms impose the highest constraints while the signature schemes that can run on these platforms are those that provide the most flexibility.

- *Terminals, tablets, HSMs, and mobile phones.* These platforms represent a broad range of devices that strike a balance between computational power, portability, and cost. Multiple processors can be chained together within the device to increase the processing throughput, but that makes the device more expensive to produce and may require a larger battery to support the increased energy needs. The CoreMark performance comparison shows that processors on these platforms perform 15 to 75 times faster than chip-cards, and already approach 10% of the overall compute throughput of servers. We classify applications that require to run on these platforms as having small constraints and signature schemes that can run on these platforms to be moderately flexible.
- *Laptops and servers.* This category represents platforms with the highest level of computational power. Such platforms are less portable but can be both horizontally and vertically scaled to have a linear increase in computation power. Applications that run on these platforms have the least constraints, while signature schemes that can only run on these platforms are the most inflexible.

3.3 Stitching the requirements into a framework

Using the identified operating constraints, we present the requirements framework in Table 2 where we assign values to each of the application use-cases discussed in Section 2, taking into account the platform that the functions are run on.

Table 2. Consolidated signing requirements of applications based on operating constraints

	Platform of execution			Execution time			Size of material		
	P1	P2	P3	$KeyGen$	$Sign$	Ver	K_s	K_p	σ
3SKey	USB token	USB token	Server	$\leq$ 1,000 ms	$\leq$ 100 ms	$\leq$ 100 ms	$\leq$ 16 Kbits	$\leq$ 16 Kbits	$\leq$ 16 Kbits
EMV-SDA	HSM	Chip-card	Tablet / Terminal	$\leq$ 1,000 ms	$\leq$ 100 ms	$\leq$ 100 ms	$\leq$ 512 Kbits	$\leq$ 2 Kbits	$\leq$ 2 Kbits
EMV-DDA	Chip-card	Chip-card	Tablet / Terminal	$\leq$ 1,000 ms	$\leq$ 100 ms	$\leq$ 100 ms	$\leq$ 16 Kbits	$\leq$ 2 Kbits	$\leq$ 2 Kbits
CA Key	HSM	HSM	Laptop / Phone	$\leq$ 1,000 ms	$\leq$ 100 ms	$\leq$ 100 ms	$\leq$ 512 Kbits	$\leq$ 512 Kbits	$>$ 512 Kbits
ICAO 9303	HSM	Chip-card	Tablet / Terminal	$\leq$ 1,000 ms	$\leq$ 100 ms	$\leq$ 100 ms	$\leq$ 512 Kbits	$\leq$ 16 Kbits	$\leq$ 2 Kbits
GSM eSIM	eUICC	eUICC	Phone	$\leq$ 1,000 ms	$\leq$ 100 ms	$\leq$ 100 ms	$\leq$ 16 Kbits	$\leq$ 16 Kbits	$\leq$ 16 Kbits
TLS server	Server	Server	Laptop / Phone	$\leq$ 1,000 ms	$\leq$ 1 ms	$\leq$ 100 ms	$>$ 512 Kbits	$\leq$ 512 Kbits	$>$ 512 Kbits
TLS client	Laptop / Phone	Laptop / Phone	Server	$\leq$ 1,000 ms	$\leq$ 100 ms	$\leq$ 1 ms	$\leq$ 512 Kbits	$\leq$ 512 Kbits	$>$ 512 Kbits
Bitcoin	Laptop / Phone	Laptop / Phone	Server	$\leq$ 100 ms	$\leq$ 100 ms	$\leq$ 1 ms	$\leq$ 512 Kbits	$\leq$ 512 Kbits	$\leq$ 16 Kbits
FIDO	USB token	USB token	Server	$\leq$ 100 ms	$\leq$ 100 ms	$\leq$ 100 ms	$\leq$ 16 Kbits	$\leq$ 16 Kbits	$\leq$ 16 Kbits
S/MIME	Laptop	Laptop	Laptop	$\leq$ 1,000 ms	$\leq$ 100 ms	$\leq$ 100 ms	$>$ 512 Kbits	$>$ 512 Kbits	$>$ 512 Kbits
PDF-AES	HSM	Laptop / Phone	Laptop / Phone	$\leq$ 1,000 ms	$\leq$ 100 ms	$\leq$ 100 ms	$\leq$ 512 Kbits	$\leq$ 512 Kbits	$>$ 512 Kbits
PDF-QES	USB token	USB token	Laptop / Phone	$\leq$ 1,000 ms	$\leq$ 100 ms	$\leq$ 100 ms	$\leq$ 16 Kbits	$\leq$ 16 Kbits	$\leq$ 16 Kbits
Code-sign	HSM	Laptop / Phone / Terminal	Laptop / Phone / Terminal	$\leq$ 1,000 ms	$\leq$ 100 ms	$\leq$ 100 ms	$\leq$ 512 Kbits	$\leq$ 512 Kbits	$\leq$ 512 Kbits

Entries colored red = Most constraints, brown = Moderate constraints, blue = Small constraints, black = Least constraints.

4 Applying the Framework

We attempt to answer the question "Will NIST's PQC Standardization yield feasible post-quantum secure digital signature drop-in replacements for all applications?" by applying the framework.

4.1 Building a Feasibility Matrix

We tabulate known performance values for the round 3 NIST PQC Standardization candidate algorithms in Table 3 using the same identified constraints. For comparison purposes, we select the algorithm that can minimally meet NIST Level 1 or AES-128 bit strength for each of the signature schemes. Execution time is obtained from [89] where possible for consistency and sizes are obtained from the round 2 submission documents. We note that [89] has already checked the results against SUPERCOP [53]. We estimate $KeyGen$ execution times from $Sign$ execution times using the ratio of CPU cycles between $KeyGen$ and $Sign$ since both functions are executed on the same platform in real-life.

Table 3. Known performance for round 3 NIST PQC Standardization candidates

Scheme	Algorithm	Execution Time (ms) [89]	Size (bits)[3]
Dilithium [26]	Dilithium II	$KeyGen = 0.18$[1]	$K_s = 22{,}400$
		$Sign = 0.82$	$K_p = 9{,}472$
		$Ver = 0.16$	$\sigma = 16{,}352$
Falcon [33]	Falcon-512	$KeyGen = 16.77$[1]	$K_s = 10{,}248$
		$Sign = 5.22$	$K_p = 7{,}176$
		$Ver = 0.05$	$\sigma = 5{,}520$
Rainbow [24]	Rainbow Ia Cyclic	$KeyGen = 0.48$[1]	$K_s = 743{,}680$
		$Sign = 0.34$	$K_p = 465{,}152$
		$Ver = 0.83$	$\sigma = 512$
GeMSS [16]	GeMSS128	$KeyGen = 13.1$[2]	$K_s = 107{,}502$
		$Sign = 188$[2]	$K_p = 2{,}817{,}504$
		$Ver = 0.03$[2]	$\sigma = 258$
Picnic [18]	Picnic-L1-FS	$KeyGen = 0.005$[1]	$K_s = 128$
		$Sign = 4.09$	$K_p = 256$
		$Ver = 3.25$	$\sigma = 272{,}256$
SPHINCS$^+$ [7]	SPHINCS$^+$-SHA256-128f-simple	$KeyGen = 2.95$[1]	$K_s = 512$
		$Sign = 93.37$	$K_p = 256$
		$Ver = 3.92$	$\sigma = 135{,}808$

[1] $KeyGen$ is 0.23 [26], 3.21 [78], 1.44 [24], 0.0013 [18], and 31.6 [7] times compared to $Sign$. .
[2] Values are obtained from [16] and with clarification from the GeMSS team.
[3] Sizes are based on NIST PQC Standardization round 2 submissions to be consistent with [89].

Since execution times carried out by [89] in Table 3 were measured on a Intel i5-8350U processor laptop, we take a rule-of-thumb assumption (refer to Table 1) that the signing functions will run 500 times slower on a chip-card, and 10 times slower on a terminal or phone. For each execution time (i.e. how fast $KeyGen$, $Sign$, Ver is on the target platform) and size (i.e. how large K_s, K_p, σ is) constraint, we score the feasibility of an application to use a signature scheme

using a linear score where "2" = within the constraint, "1" = within 200% of the constraint, and "0" otherwise. We choose to include a 200% constraint window as a sensitivity parameter to take into account i) errors in the rule-of-thumb, ii) variations to performance results, and iii) possible algorithmic improvements in the future. A feasibility score of 11 or 12 points would likely represent a viable fit between application and signature scheme, while lower scores may point to less feasibility and identify areas that can be optimized. Table 4 shows the feasibility matrix when applying each of the signature schemes to each application.

Table 4. Feasibility Matrix of Algorithm against Application

	PQC round 3 finalists			PQC round 3 alternates		
	Dilithium	Falcon	Rainbow	GeMSS	Picnic	SPHINCS$^+$
3SKey	9	8	7	4	8	7
EMV-SDA	8	9	9	8	10	8
EMV-DDA	5	4	7	4	8	7
CA Key	12	12	11	8	12	10
ICAO 9303	10	10	9	8	10	8
GSM eSIM	9	10	7	4	8	7
TLS server	12	10	12	8	10	10
TLS client	12	12	11	8	10	8
Bitcoin	12	11	11	7	8	6
FIDO	9	8	5	4	8	6
S/MIME	12	12	12	10	12	12
PDF-AES	12	12	11	8	12	10
PDF-QES	9	8	7	4	8	7
Code-sign	12	12	11	10	12	10
Mean score	10.21	9.85	9.29	6.78	9.71	8.28
Score $\geq$ 11	7	6	7	0	4	1
Worst score	5	4	5	4	8	6

From Table 4, Dilithium scored the best mean score of 10.21, followed by Falcon (9.85), Picnic (9.71), and Rainbow (9.29). When examining the number of applications feasibly supported by the algorithms (i.e. having a score of 11 and above), each of the finalists score a minimum of 11 for between 6 and 7 applications, as compared to the alternates whose best-performing algorithm, Picnic, scored 11 and above only for 4 applications. There are 7 applications (i.e. 3SKey, EMV-SDA, EMV-DDA, ICAO 9303, GSM eSIM, FIDO, PDF-QES) that do not have any algorithm scoring 11 and above. The commonality is that they use chip-card platforms for P1 and/or P2 where all algorithms score poorly in meeting the *Sign* execution time and the σ size requirements[3]. In looking at worst scores, Picnic fared the best amongst the algorithms, scoring a minimum of 8 across all the applications compared to the other algorithms whose worst scores are between 4 to 6.

[3] We note that the estimated *KeyGen* execution times did not materially influence the eventual result.

We validate our findings by looking at implementations in constrained platforms. PQC implementations on SoCs (System-on-chips), ASICs (Application Specific Integrated Circuits), and FPGAs (Field Program-mable Gate Arrays) are active fields of research [73, 47, 9, 23, 70] and have yielded promising results. But this is not as prevalent for chip-cards. A search on related research implementations of PQC algorithms on chip-cards yielded no signature examples and two encryption examples [14, 5]. Notably, both encryption implementations worked on key sizes that are lower than 128-bit security and neither of them included active defenses against side-channel attacks. Hence to answer the question:

Q: Will NIST's PQC Standardization yield feasible post-quantum secure digital signature drop-in replacements for all applications?
A: Yes for 7 out of 14 application domains, so many challenges still remain.

4.2 Future Directions

We list the following research areas to help close the gap:

1. *Chip-card research.* Benchmarking the actual processing throughput of the chip-card. In this paper, we have used a factor of 500 to estimate the difference in speed between a chip-card and a laptop. This method is neither rigorous nor accurate. Work on measuring the NIST PQC Standardization round 3 candidates on chip-card platforms will be an important part of PQC research.
2. *Algorithm research.*
   - *NIST PQC round 3 finalists.* Based on our framework, all 3 finalists can feasibly support 6 to 7 applications of the 14 identified applications. Beyond TLS which has been well researched by [89], we urge the community to validate their feasibility by deploying Dilithium, Falcon, and Rainbow on these applications, and sharing the results.
   - *Picnic.* When referring to Table 4 and focusing only on the 7 chip-card based applications, we note that Picnic stands out as the highest-scoring algorithm. Besides being the algorithm with the highest worst score overall, Picnic has only lost points in all applications due to its inability to meet the execution time requirements of $Sign$ and the size requirements of σ. We expect that improvements in these 2 areas will make Picnic a strong signature candidate for chip-card applications.

5 Conclusion

We have done a comprehensive survey of 14 real-world applications that use digital signatures across 4 broad industries. We constructed a requirements framework and assigned appropriate operating constraint values for the execution time of $KeyGen$, $Sign$, and Ver as well as the sizes for K_s,K_p, and σ. Upon applying the framework to build a feasibility matrix, we showed that applications that use chip-card based platforms to carry out the $Sign$ function or to

output the signature σ may not have a suitable post-quantum digital signature replacement, and this warrants more research.

Acknowledgements

This project is partially supported by the Ministry of Education, Singapore, under its MOE AcRF Tier 2 grant (MOE2018-T2-1-111).

References

1. Adobe: Adobe DC Digital Signatures Guide - Supported Standards. Online: https://www.adobe.com/devnet-docs/acrobatetk/tools/DigSigDC/standards.html [accessed: April 2021] (2018)
2. Ajtai, M.: Generating hard instances of lattice problems. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 99–108. ACM (1996)
3. Akkar, M.L., Courtois, N.T., Duteuil, R., Goubin, L.: A fast and secure implementation of sflash. In: International Workshop on Public Key Cryptography. pp. 267–278. Springer (2003)
4. Alagic, G., Alagic, G., Alperin-Sheriff, J., Apon, D., Cooper, D., Dang, Q., Liu, Y.K., Miller, C., Moody, D., Peralta, R., et al.: Status report on the first round of the NIST post-quantum cryptography standardization process. US Department of Commerce, National Institute of Standards and Technology (2019)
5. Albrecht, M.R., Hanser, C., Hoeller, A., Pöppelmann, T., Virdia, F., Wallner, A.: Implementing rlwe-based schemes using an rsa co-processor. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 169–208 (2019)
6. Augot, D., Batina, L., Bernstein, D.J., Bos, J., Buchmann, J., Castryck, W., Dunkelman, O., Güuneysu, T., Gueron, S., Hülsing, A., Lange, T., Mohamed, M.S.E., Rechberger, C., Schwabe, P., Sendrier, N., Vercauteren, F., Yang, B.Y.: Initial recommendations of long-term secure post-quantum systems. Online: http://pqcrypto.eu.org/docs/initial-recommendations.pdf [accessed: April 2021] (2015)
7. Aumasson, J.P., Bernstein, D.J., Dobraunig, C., Eichlseder, M., Fluhrer, S., Gazdag, S.L., Hülsing, A., Kampanakis, P., Kölbl, S., Lange, T., Lauridsen, M.M., Mendel, F., Niederhagen, R., Rechberger, C., Rijneveld, J., Schwabe, P.: SPHINCS$^+$ Submission to the NIST post-quantum project. Online: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/submissions/SPHINCS-Round2.zip [accessed: April 2021] (2019)
8. Bai, S., Galbraith, S.D.: An improved compression technique for signatures based on learning with errors. In: Cryptographers' Track at the RSA Conference. pp. 28–47. Springer (2014)
9. Banerjee, U., Ukyab, T.S., Chandrakasan, A.P.: Sapphire: A configurable crypto-processor for post-quantum lattice-based protocols. arXiv preprint arXiv:1910.07557 (2019)
10. Barker, E.: SP 800-57 part 1 rev. 5 Recommendation for key management part 1: General. NIST special publication **800**, 57 (2020)
11. Barker, W., Polk, W., Souppaya, M.: Getting ready for post-quantum cryptography: Explore challenges associated with adoption and use of post-quantum cryptographic algorithms. NIST Cybersecurity White Paper (2021)

12. Bennett, C.H., Bernstein, E., Brassard, G., Vazirani, U.: Strengths and weaknesses of quantum computing. SIAM journal on Computing **26**(5), 1510–1523 (1997)
13. Bernstein, D.J., Hopwood, D., Hülsing, A., Lange, T., Niederhagen, R., Papachristodoulou, L., Schneider, M., Schwabe, P., Wilcox-O'Hearn, Z.: SPHINCS: practical stateless hash-based signatures. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 368–397. Springer (2015)
14. Boorghany, A., Sarmadi, S.B., Jalili, R.: On constrained implementation of lattice-based cryptographic primitives and schemes on smart cards. ACM Transactions on Embedded Computing Systems (TECS) **14**(3), 1–25 (2015)
15. Buchmann, J., Dahmen, E., Hülsing, A.: XMSS-a practical forward secure signature scheme based on minimal security assumptions. In: International Workshop on Post-Quantum Cryptography. pp. 117–129. Springer (2011)
16. Casanova, A., Faugere, J.C., Macario-Rat, G., Patarin, J., Perret, L., Ryckeghem, J.: Gemss: A great multivariate short signature. Online: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/submissions/GeMSS-Round2.zip [accessed: April 2021] (2019)
17. Chalkias, K., Brown, J., Hearn, M., Lillehagen, T., Nitto, I., Schroeter, T.: Blockchained post-quantum signatures. In: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). pp. 1196–1203. IEEE (2018)
18. Chase, M., Derler, D., Goldfeder, S., Katz, J., Kolesnikov, V., Orlandi, C., Ramacher, S., Rechberger, C., Slamanig, D., Wang, X., Zaverucha, G.: The picnic digital signature algorithm: Update for round 2 (2019)
19. Chase, M., Derler, D., Goldfeder, S., Orlandi, C., Ramacher, S., Rechberger, C., Slamanig, D., Zaverucha, G.: Post-quantum zero-knowledge and signatures from symmetric-key primitives. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. pp. 1825–1842. ACM (2017)
20. Chen, L., Chen, L., Jordan, S., Liu, Y.K., Moody, D., Peralta, R., Perlner, R., Smith-Tone, D.: NISTIR 8105: Report on post-quantum cryptography. US Department of Commerce, National Institute of Standards and Technology (2016)
21. Cooper, D.A., Apon, D.C., Dang, Q.H., Davidson, M.S., Dworkin, M.J., Miller, C.A.: Recommendation for stateful hash-based signature schemes. NIST Special Publication **800**, 208 (2020)
22. Courtois, N., Goubin, L., Meier, W., Tacier, J.D.: Solving underdefined systems of multivariate quadratic equations. In: International Workshop on Public Key Cryptography. pp. 211–227. Springer (2002)
23. Dang, V.B., Farahmand, F., Andrzejczak, M., Mohajerani, K., Nguyen, D.T., Gaj, K.: Implementation and benchmarking of round 2 candidates in the nist post-quantum cryptography standardization process using hardware and software/hardware co-design approaches. Cryptology ePrint Archive: Report 2020/795 (2020)
24. Ding, J., Chen, M.S., Petzoldt, A., Schmidt, D., Yang, B.Y.: Rainbow-algorithm specification and documentation: The 2nd round proposal. Online: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/submissions/Rainbow-Round2.zip [accessed: April 2021] (2019)
25. Dubois, V., Fouque, P.A., Shamir, A., Stern, J.: Practical cryptanalysis of SFLASH. In: Annual International Cryptology Conference. pp. 1–12. Springer (2007)

26. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS–Dilithium: Algorithm Specification and Supporting Documentation. Round-2 submission to the NIST PQC project (2019)
27. EEMBC: CoreMark: An EEMBC Benchark. Online: https://www.eembc.org/coremark/scores.php [accessed: April 2021] (2020)
28. EMVCo: EMV Integrated Circuit Card Specifications for Payment Systems Book 2 Security and Key Management Version 4.3 (2011)
29. ETSI: ETSI TS 102 778-1 V1.1.1 Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 1: PAdES Overview - a framework document for PAdES. Online: https://www.etsi.org/deliver/etsi_ts/102700_102799/ 10277801/01.01.01 _60/ts_10277801v010101p.pdf [accessed: April 2021] (2009)
30. ETSI: Quantum Safe Cryptography; Case Studies and Deployment Scenarios ETSI GR QSC 003 V1.1.1. Online: https://www.etsi.org/deliver/etsi_gr/QSC/001_099/003 /01.01.01_60/ gr_QSC003v010101p.pdf [accessed: April 2021] (2017)
31. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Conference on the Theory and Application of Cryptographic Techniques. pp. 186–194. Springer (1986)
32. FIPS, P.: 186-4: Federal information processing standards publication. Digital Signature Standard (DSS). Information Technology Laboratory, National Institute of Standards and Technology (NIST), Gaithersburg, MD (2013)
33. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU Specifications v1.1. Online: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/submissions/Falcon-Round2.zip [accessed: April 2021] (2019)
34. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the fortieth annual ACM symposium on Theory of computing. pp. 197–206. ACM (2008)
35. Giacomelli, I., Madsen, J., Orlandi, C.: Zkboo: Faster zero-knowledge for boolean circuits. In: 25th {USENIX} Security Symposium ({USENIX} Security 16). pp. 1069–1083 (2016)
36. Goldreich, O.: Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press (2009)
37. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: Annual International Cryptology Conference. pp. 112–131. Springer (1997)
38. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM Journal on computing **18**(1), 186–208 (1989)
39. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing **17**(2), 281–308 (1988)
40. Grover, L.K.: Quantum mechanics helps in searching for a needle in a haystack. Physical review letters **79**(2), 325 (1997)
41. GSMA: eSIM whitepaper: The what and how of remote sim provisioning. Online: https://www.gsma.com/esim/wp-content/uploads/2018/06/eSIM-Whitepaper-v4.11.pdf [accessed: April 2021] (2018)

42. Güneysu, T., Lyubashevsky, V., Pöppelmann, T.: Practical lattice-based cryptography: A signature scheme for embedded systems. In: International Workshop on Cryptographic Hardware and Embedded Systems. pp. 530–547. Springer (2012)
43. Hanke, T., Movahedi, M., Williams, D.: Dfinity technology overview series, consensus system. arXiv preprint arXiv:1805.04548 (2018)
44. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: International Algorithmic Number Theory Symposium. pp. 267–288. Springer (1998)
45. Hülsing, A.: W-OTS+–shorter signatures for hash-based signature schemes. In: International Conference on Cryptology in Africa. pp. 173–188. Springer (2013)
46. Hülsing, A., Butin, D., Gazdag, S., Rijneveld, J., Mohaisen, A.: XMSS: eXtended Merkle signature scheme. Online: https://tools.ietf.org/html/rfc8391 [accessed: April 2021] (2018)
47. Hülsing, A., Rijneveld, J., Schwabe, P.: Armed sphincs. In: Public-Key Cryptography–PKC 2016, pp. 446–470. Springer (2016)
48. ICAO: Doc 9303: Machine Readable Travel Documents. Online: https://www.icao.int/publications/pages/publication.aspx?docnum=9303 [accessed: April 2021] (2015)
49. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. In: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing. pp. 21–30. ACM (2007)
50. ISO: ISO 32000-1:2008 Document management — Portable document format — Part 1: PDF 1.7. Online: https://www.iso.org/standard/51502.html [accessed: April 2021] (2013)
51. ISO: ISO/IEC 7816-4:2013 identification cards — integrated circuit cards — part 4: Organization, security and commands for interchange. Online: https://www.iso.org/standard/54550.html [accessed: April 2021] (2013)
52. ITU-T, X.: Information Technology–Open Systems Interconnection–The Directory: Public-Key and Attribute Certificate Frameworks. Online: https://www.itu.int/rec/T-REC-X.509-201910-I/en [accessed: April 2021] (2019)
53. J., B.D.: ebacs: Ecrypt benchmarking of cryptographic systems. Online: https://bench.cr.yp.to/primitives-sign.html [accessed: April 2021] (2019)
54. Kiayias, A., Russell, A., David, B., Oliynykov, R.: Ouroboros: A provably secure proof-of-stake blockchain protocol. In: Annual International Cryptology Conference. pp. 357–388. Springer (2017)
55. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 206–222. Springer (1999)
56. Koblitz, N.: Elliptic curve cryptosystems. Mathematics of computation **48**(177), 203–209 (1987)
57. Lamport, L.: Constructing digital signatures from a one-way function. Tech. rep., Technical Report CSL-98, SRI International Palo Alto (1979)
58. Langner, R.: Stuxnet: Dissecting a cyberwarfare weapon. IEEE Security & Privacy **9**(3), 49–51 (2011)
59. Lee, C.C., Tan, T.G., Sharma, V., Zhou, J.: Quantum computing threat modelling on a generic cps setup. In: International Conference on Applied Cryptography and Network Security. pp. 171–190. Springer (2021)
60. Leighton, F.T., Micali, S.: Large provably fast and secure digital signature schemes based on secure hash functions (1995), uS Patent 5,432,852

61. Lyubashevsky, V.: Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 598–616. Springer (2009)
62. Machani, S., Philpott, R., Srinivas, S., Kemp, J., Hodges, J.: FIDO UAF Architectural Overview. Online: https://fidoalliance.org/specs/fido-uaf-v1.1-ps-20170202/fido-uaf-overview-v1.1-ps-20170202.pdf [accessed: April 2021] (2017)
63. Manulis, M., Bridges, C.P., Harrison, R., Sekar, V., Davis, A.: Cyber security in new space. International Journal of Information Security **20**(3), 287–311 (2021)
64. Matsumoto, T., Imai, H.: Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In: Workshop on the Theory and Application of of Cryptographic Techniques. pp. 419–453. Springer (1988)
65. McEliece, R.J.: A public-key cryptosystem based on algebraic. Coding Thv **4244**, 114–116 (1978)
66. Merkle, R.C.: A certified digital signature. In: Conference on the Theory and Application of Cryptology. pp. 218–238. Springer (1989)
67. Miller, R.B.: Response time in man-computer conversational transactions. In: Proceedings of the December 9-11, 1968, fall joint computer conference, part I. pp. 267–277 (1968)
68. Moody, D.: The 2nd round of the nist pqc standardization process. Online: https://csrc.nist.gov/CSRC/media/Presentations/the-2nd-round-of-the-nist-pqc-standardization-proc/images-media/moody-opening-remarks.pdf [accessed: April 2021] (2019)
69. Moody, D.: NIST PQC Standardization Update - Round 2 and Beyond. Online: https://csrc.nist.gov/CSRC/media/Presentations/pqc-update-round-2-and-beyond/images-media/pqcrypto-sept2020-moody.pdf [accessed: April 2021] (2020)
70. multiple: Post-quantum crypto library for the arm cortex-m4. Online: https://github.com/mupq/pqm4 [accessed: April 2021] (2020)
71. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Online: https://bitcoin.org/bitcoin.pdf [accessed: April 2021] (2008)
72. NIST: Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process. Online: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf [accessed: April 2021] (2016)
73. Oder, T., Pöppelmann, T., Güneysu, T.: Beyond ecdsa and rsa: Lattice-based digital signatures on constrained devices. In: 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC). pp. 1–6. IEEE (2014)
74. Patarin, J.: Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88. In: Annual International Cryptology Conference. pp. 248–261. Springer (1995)
75. Patarin, J.: Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 33–48. Springer (1996)
76. Patarin, J.: The oil and vinegar signature scheme. In: Dagstuhl Workshop on Cryptography September, 1997 (1997)
77. Patarin, J., Courtois, N., Goubin, L.: Quartz, 128-bit long digital signatures. In: Cryptographers' Track at the RSA Conference. pp. 282–297. Springer (2001)
78. Pornin, T.: New Efficient, Constant-Time Implementations of Falcon. Cryptology ePrint Archive, Report 2019/893 (2019), https://eprint.iacr.org/2019/893
79. Proos, J., Zalka, C.: Shor's discrete logarithm quantum algorithm for elliptic curves. arXiv preprint quant-ph/0301141 (2003)

80. Qu, M.: SEC 2: Recommended elliptic curve domain parameters. Certicom Res., Mississauga, ON, Canada, Tech. Rep. SEC2-Ver-0.6 (1999)
81. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM (JACM) **56**(6), 34 (2009)
82. Rescorla, E.: The transport layer security (TLS) protocol version 1.2. Online: https://tools.ietf.org/html/rfc5246 [accessed: April 2021] (2008)
83. Rescorla, E.: The transport layer security (TLS) protocol version 1.3. Online: https://tools.ietf.org/html/rfc8446 [accessed: April 2021] (2018)
84. Reyzin, L., Reyzin, N.: Better than BiBa: Short one-time signatures with fast signing and verifying. In: Australasian Conference on Information Security and Privacy. pp. 144–153. Springer (2002)
85. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM **21**(2), 120–126 (1978)
86. Rostovtsev, A., Stolbunov, A.: Public-key cryptosystem based on isogenies. IACR Cryptology ePrint Archive **2006**, 145 (2006)
87. Schaad, J., Cellars, A., Ramsdell, B., Turner, S.: Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification. Online: https://tools.ietf.org/html/rfc8551 [accessed: April 2021] (2019)
88. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review **41**(2), 303–332 (1999)
89. Sikeridis, D., Kampanakis, P., Devetsikiotis, M.: Post-Quantum Authentication in TLS 1.3: A Performance Study. In: 27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020. The Internet Society (2020)
90. SWIFT: How much do you pay for your PKI solution? Online: https://www.swift.com/file/ 29886/download?token=ic6vj_vD [accessed: April 2021] (2016)
91. Takahashi, Y., Kunihiro, N.: A quantum circuit for shor's factoring algorithm using 2n+2 qubits. Quantum Information & Computation **6**(2), 184–192 (2006)
92. Tan, T.G., Zhou, J.: Layering quantum-resistance into classical digital signature algorithms. In: International Conference on Information Security. pp. 26–41. Springer (2021)
93. VASCO: VASCO Announces Bankruptcy Filing by DigiNotar B.V. Online: https://web.archive.org/web/20110923180445/http://www.vasco.com/company/press_room/news_archive/2011/news_vasco_announces_bankruptcy_filing_by_diginotar_bv.aspx [accessed: April 2021] (2011)
94. Wallden, P., Kashefi, E.: Cyber security in the quantum era. Commun. ACM **62**(4), 120 (2019)

A Digital Signing Background

Applications use digital signatures to achieve data integrity, user authenticity, and message non-repudiation. Between Alice and Bob, data integrity ensures that any message sent from Alice to Bob can be verified by Bob if the message received was unmodified, or had been modified in transit. User authenticity ensures that Bob is able to ascertain if Alice is who she claims she is. Message non-repudiation ensures that Bob is able to prove that the message originated from Alice, and Alice is unable to deny that proof. While data integrity and

user authenticity can also be achieved using other security primitives such as message authentication codes (MAC), only digital signatures can provide the non-repudiation capability that is needed in many business applications.

A.1 Digital Signature Scheme

We define a digital signature scheme as a triple of polynomial-time signing functions $KeyGen$, $Sign$, Ver with the following parameters:

$KeyGen(1^n) \Rightarrow (K_s, K_p)$ takes in a security parameter 1^n which defines the cryptographic key strength of n, and outputs a secret key K_s and corresponding public key K_p.

$Sign(M, K_s) \Rightarrow (\sigma)$ takes in a message M and the secret key K_s, and outputs a signature σ.

$Ver(M, K_p, \sigma) \Rightarrow (result)$ takes in a message M, the public key K_p and signature σ, and outputs accept if and only if σ is a valid signature generated by $Sign(M, K_s)$.

A digital signature scheme is deemed secure if it is proven to be existential unforgeable under chosen message attack (EUF-CMA) [39]. The EUF-CMA experiment, described below, between Alice and adversary Mallory seeks to prove that Mallory will not be any closer to forging a signature despite multiple interactions with Alice.

1. Alice performs $KeyGen$ and sends the public key K_p to Mallory.
2. Mallory can choose a message M_i and ask Alice to sign the message.
3. Alice signs the message M_i using $Sign(M_i, K_s)$ and returns the signature σ_i to Mallory. Step 2) and 3) are repeated multiple times.
4. At the end of the experiment, Mallory has to output a message M' that is not within the set of messages M_i requested in Step 2), and also a signature σ' that will return accept when $Ver(M', K_p, \sigma')$ is called.

A.2 Threat from Quantum Computers

Quantum computers work on the concept of a quantum bit (qubit) where each qubit can exist in a state of superposition between states 0 and 1 during computation, until the qubit is finally accessed. During computation, quantum algorithms use quantum gates or circuits to make qubits interact by the principles of superposition, interference, and entanglement to achieve computational speed-ups that are not possible in classical computers. Cybersecurity threats posed by quantum computers may come in different forms [94, 59] but they all stem from the vulnerability caused by Shor's algorithm [88] running on a quantum computer. NIST has also highlighted the need to protect public key cryptography against Shor's algorithm in their update [69] and white paper [11].

The security of the RSA cryptosystem is based on the hard problem of factorizing a very large modulus m which is a multiple of 2 prime factors p and q [85]. In order to cryptanalyze an RSA key, the prime factors p and q need to be recovered, given the modulus m. The classical factorization algorithm is described in Algorithm 1.

Shor's algorithm [88] comes into play only in step 12 of Algorithm 1 where no classical or probabilistic polynomial-time algorithm exists for the order finding

Algorithm 1: Factorization of RSA.

```
1  begin
2      m ← modulus;
3      if m is even then
4          p = 2; q = m/2; # solved
5      else
6          while not solved do
7              a ← random;
8              Compute g = GCD(a, m) ;
9              if g > 1 then
10                 p = g; q = m/p; # solved
11             else
12                 Search for r where a^r ≡ 1 mod m ;
13                 if r is even then
14                     Compute x ≡ a^(r/2) mod m ;
15                     if 1 < x < m − 1 then
16                         p = GCD(x + 1, m) ;
17                         q = GCD(x − 1, m); #solved
18                     end
19                 end
20             end
21         end
22     end
23 end
```

problem. Through the use of phase estimation and quantum fourier transform, Shor's algorithm achieves polynomial-time complexity to find the order r for $a^r \equiv 1 \; mod \; m$ where m is the modulus and a is a random less than $\frac{m}{2}$. This effectively reduces the time required to find an RSA private key from billions of years to a matter of hours, thus breaking any EUF-CMA security assumption. The number of qubits needed on a quantum computer to break the RSA cryptosystem is estimated at $2n + 2$ [91] where n is the size of the modulus in bits. This means that there needs to be a 4,000+ qubit Quantum computer to break an RSA-2048 signature. Shor's algorithm can also be adapted to solve the discrete logarithm problem and the number of qubits to break ECDSA is "roughly" $6n$ [79], or a 1,500+ qubits Quantum computer to break an ECDSA-256 bit signature.

B Post-Quantum Digital Signatures

We briefly cover the various families of post-quantum digital signature algorithms being evaluated for PQC below, and highlight where the 6 remaining candidate algorithms in the NIST PQC round 3 are.

B.1 Lattice-based Cryptography

Lattice-based cryptography has by far gained the most attention as the potential PQC candidate to address the threat of quantum computers. We believe

that the interest is due to the flexibility of the lattice structure to support both encryption and digital signing, as well as the reasonable key and signature sizes of 10-20Kbits for a 128-bit security strength. Of all the categories classified under the NIST PQC Standardization, only lattice-based cryptography has round 3 candidates in both key exchange and digital signing.

Lattice-based cryptography is a class of cryptographic primitives built on a multi-dimensional lattice structure and first used by Ajtai in 1996 [2]. Ajtai presented a cryptographic construction using lattices based on the short integer solution (SIS) problem and showed it secure in the average case if the shortest vector problem (SVP) was hard in the worst case. Goldreich, Goldwasser, and Halevi (GGH) [37] introduced a more practical variant based on the lattice-reduction or closest vector problem. GGH eventually gave rise to other SVP lattice algorithms such as Nth degree Truncated Polynomial Ring Unit (NTRU) [44]. Regev [81] used a different hard problem based on learning with errors (LWE) problem and showed that it was similarly secure if lattice problems were hard in the worst case.

Using SVP and its variants, Gentry, Peikert, and Vaikuntanathan (GPV) [34] formalized a provably secure (over classical and quantum oracle models) hash-and-sign trapdoor approach to digital signing over lattices. The trapdoor, in this case, refers to the short-basis (or private key) of the lattice that is used to generate the short vector (or signature) which points to the hash of the message on the lattice. A different class of non-trapdoor signatures over lattices using the Fiat-Shamir's heuristic [31] was proposed by Lyubashevsky [61].

There are 2 lattice-based candidate signature algorithms in round 3 of the NIST PQC Standardization as finalists:

- *Dilithium* is part of Crystals (Cryptographic Suite for Algebraic Lattices, www.pq-crystals.org) which includes both a key-exchange algorithm, Kyber, and a digital signature algorithm, Dilithium. Dilithium uses Fiat-Shamir's signing scheme [31] on a module-LWE problem. The designers deliberately wanted to reduce the public key and signature sizes by improving on the work by Bai and Galbraith [8] and added the concept of "hints" [42] to do so.
- *Falcon* (Fast-Fourier Lattice-based Compact Signatures over NTRU, www.falcon-sign.info) is based on GPV [34] by applying the SIS problem over NTRU lattices [44]. For the trapdoor sampling, the designers used fast Fourier sampling to improve signing time while achieving a shorter short vector.

B.2 Code-based Cryptography

Code-based cryptography was proposed by McEliece [65] in 1978 which described an asymmetric key cryptographic system based on the hardness of decoding a generic linear code, an NP-hard problem. A linear code is essentially a form of error-correcting codes with linear combination properties. The private key in code-based cryptography is typically a code C, which has the ability to correct t errors. When sending a message, the sender will encode the message with the public key and include t errors within the encoding, and the receiver

with code C will be able to decode the message while accurately correcting the errors. Typical key sizes for code-based cryptography exceed 1Mbits to achieve 128-bit strength in security.

There were 2 digital signature proposals in the NIST PQC Standardization round 1 but both had attacks published in the subsequent public consultation and neither managed to move to round 2 [68].

B.3 Multi-Variate Cryptography

Multi-variate cryptography is a broad class of cryptographic techniques encompassing algorithms that rely on the difficulty of solving n unknowns (or variables) within p multivariate polynomial equations. When performing digital signing, the set of p equations (with s unknowns) is the public key while the appropriate values of n is the signature. During signing, the signer is in possession of the private key which consists of 2 affine transformations and a carefully crafted set of polynomial equations that allow for the trapdoor function computation of n from the hash of the message. The verifier can apply n into the p equations to verify that the output of the p equations corresponds to the hash of the message that is signed. While it sounds intuitively NP-hard, much of the security lies with the underlying multi-variate scheme, the choice of the parameters s and p, and the design of the trapdoor function needed to support public key cryptography [22].

One of the earliest multi-variate cryptographic constructs was by Matsumoto and Imai [64] who proposed C* in 1988. It was subsequently broken by Patarin [74] who used the general principle to introduce Hidden Field Equations (HFE) [75] and Balanced Oil and Vinegar [76]. Kipnis, Pararin, and Goubin then introduced Unbalanced Oil and Vinegar (UOV) [55] which is the basis for Rainbow [24]. The attractiveness of multi-variate cryptography as a PQC candidate lies in its promise to have a much smaller key size. Sflash [3], a C* variant, was included in European Consortium NESSIE Project in 2003 due to its ability to fit in an 8-bit smartcard. but was broken in 2007 [25].

There are 2 multi-variate based candidate signature algorithms in round 3 of NIST PQC Standardization:

- *Rainbow* was first introduced in 2005 [24] as a generalization of UOV to allow for multiple layers, each with different parameters chosen. Additional layers could improve the security strength of the overall signature construct, but impacts the efficiency and resources needed by the proving and verifying entities. Rainbow is a NIST PQC Standardization round 3 finalist.
- *GeMSS* (A Great Multivariate Short Signature, www-polsys.lip6.fr/Links/NIST/GeMSS.html) is based on HFE [75] and enhances the work from Quartz [77]. Quartz already generates very small signatures of 128 bits, and the designers designed GeMSS as a faster variant of Quartz with better signing efficiency. GeMSS is a NIST PQC round 3 alternate candidate.

B.4 Hash-based Cryptography

Hash-based digital signature was first introduced by Lamport in 1979 [57]. The concept relies on the difficulty of finding the pre-image of the hash function which is essentially the preimage resistance property in good one-way hash functions. Hash-based digital signatures are notorious for their large signature sizes, and a limited key lifetime since the pre-image (when used as the secret key) can only be used once. On the other hand, interests remain high to use hash-based cryptography as a post-quantum algorithm due to its proven resistance against quantum computers. Using a quantum computer, an adversary can only maximally [12] achieve quadratic speedup when using Grover's algorithm [40] to carry out a brute-force search to find the pre-image of a hash.

There have been several improvements to Lamport's one-time signature (OTS) through variations in the use of Merkle trees [45, 66, 46] to extend the function of the secret key into a multi-use derivation secret, as well as to reduce the size of the signature. This formed the basis of stateful hash-based signatures [15, 60]. To achieve a finite number of stateless signatures, Reyzin's few-time signature Hash-to-Obtain-Random-Subset (HORS) [84] is used to transform an OTS into an N-time signature scheme where the same private key can be used to securely sign N signatures. As each signature reveals a portion of the private key, there is a security degradation from the $N+1$ signature onwards. SPHINCS by Bernstein et. al. [13] builds on Goldreich's [36] stateless hyper tree construct to obtain more private signing keys and uses HORST (adapted from HORS with trees) as the leaves of the trees to increase the number of signatures per key.

SPHINCS$^+$ (Stateless Practical Hash-based Incredibly Nice Cryptographic Signatures, www.sphincs.org) is a stateless hash-based signature. It improves on SPHINCS' [13] HORST design to obtain faster and small signatures. SPHINCS$^+$ is the only hash-based digital signing candidate in round 3 of NIST PQC Standardization as an alternate candidate.

B.5 Isogeny-based Cryptography

Isogeny refers to the mathematical mapping or morphism between 2 mathematical structures. In the case of [86], the public key system is based on the difficulty of finding the isogeny of 2 elliptic curves. In a mapping over an elliptic curve E where the secret isogeny ϕ is mapped to $E/<P>$, and the secret isogeny ψ is mapped to $E/<Q>$, revealing E, $E/<P>$, and $E/<Q>$ does not allow the adversary to know ϕ or ψ. Hence, similar to a Diffie-Hellman key exchange, two communicating parties can each generate a respective secret ϕ and ψ, and arrive at $E/<P, Q>$ to form a shared-secret securely. The most promising advantage of isogeny-based cryptography is the size of the keys which is relatively small compared to the other schemes. As the isogenies are based on elliptic curves, key sizes range from 768bits to 1024bits for an equivalent 128-bit strength in security. Unfortunately, no isogeny-based digital signature schemes were submitted for the NIST PQC Standardization. Hence, isogeny-based cryptography as with code-based cryptography, is only considered for PQC key-exchange.

B.6 Zero-Knowledge

Zero-knowledge proofs have their origins in 1985 when Goldwasser, Micali, and Rackoff [38] defined the concept of zero-knowledge as proof that "convey no additional knowledge other than the correctness of the proposition".

Non-interactive zero-knowledge proofs of knowledge constructions such as MPC-in-the-head [49] and ZKBoo [35] rely on multiparty computation (MPC) with collision-resistant one-way functions, which could be in the form of strong hash functions (e.g. SHA2-256, SHA3-384) or symmetric key encryption functions (e.g. AES-256) to complete the proof. Informally, the MPC zero-knowledge proof works by the prover splitting the secret into multiple shares (e.g. using exclusive-OR) and committing to the hash of each share. When the verifier challenges the prover on a subset of the shares, the prover is able to produce a "view" for the subset without revealing the actual values. Repeated challenges will increase the level of assurance in the proof.

Picnic (https://microsoft.github.io/picnic) is is the only zero-knowledge proof digital signature scheme in round 3 of NIST's PQC Standardization as an alternate candidate. It uses ZKB++ [19], a variant of ZKBoo, as the zero-knowledge proof where the underlying MPC circuit is the LowMC encryption scheme and the hash function used is SHAKE (a SHA-3 derived function). The Picnic signature scheme is made non-interactive through the use of either the Fiat-Shamir or the Unruh transform.