Signing Information in the Quantum Era

K. Longmate,[1] E.M. Ball,[1] E. Dable-Heath,[2] and R.J. Young[1]
[1)]Physics Department, Lancaster University, Lancaster LA1 4YB, United Kingdom
[2)]Electrical and Electronic Engineering, Imperial College London, South Kensington, London SW7 2BU, United Kingdom

(Dated: November 10, 2020)

Signatures are primarily used as a mark of authenticity, to demonstrate that the sender of a message is who they claim to be. In the current digital age, signatures underpin trust in the vast majority of information that we exchange, particularly on public networks such as the internet. However, schemes for signing digital information which are based on assumptions of computational complexity are facing challenges from advances in mathematics, the capability of computers, and the advent of the quantum era. Here we present a review of digital signature schemes, looking at their origins and where they are under threat. Next, we introduce post-quantum digital schemes, which are being developed with the specific intent of mitigating against threats from quantum algorithms whilst still relying on digital processes and infrastructure. Finally, we review schemes for signing information carried on quantum channels, which promise provable security metrics. Signatures were invented as a practical means of authenticating communications and it is important that the practicality of novel signature schemes is considered carefully, which is kept as a common theme of interest throughout this review.

CONTENTS

I. INTRODUCTION

Physical signatures are marks made to identify or authenticate the creator of a message or artifact. Their precise origins are lost to history, but they are associated with some of the earliest records of pictographic scripts, dating back at least 5 millennia[1]. The information and telecommunications revolution in the second half of the 20th century would not have happened without a practical means to authenticate messages, which led to the invention of digital signature schemes.

Schemes for signing digital information are a direct, albeit stronger, analogue to physical signatures; they seek to ensure (i) authenticity of any claim regarding a message sender's identity, (ii) that the message has not been altered by any parties since the signing, and (iii) that the sender cannot refute that it was indeed them who signed the message. In chapter 2, we review the origins of digital signatures, and explore some of the important bases for signatures in the modern world. From here, we review their applications and vulnerabilities associated with assumptions made in their foundations, whilst discussing a select few ways in which signatures have evolved for standardisation and specific use cases.

Post-quantum cryptography focuses on building classical algorithms whose security is resistant to known capabilities of quantum algorithms. Post-quantum signature schemes build upon early work in digital signatures. In chapter 3 we review progress made in this field and look closely at the resources required to implement these emerging algorithms.

It has been shown that algorithms using information encodes on quantum states can be used for secure communication protocols that are not dependent upon unproven assumptions, but instead are provably secure within the laws of physics itself. Chapter 4 discusses the application of quantum information and communications to signature schemes.

II. CLASSICAL SIGNATURES

The pursuit of secure digital signature schemes was of great importance in 20th-century cryptographic research. Digital signatures are considered to be a cryptographic primitive with widespread application and use, with legal precedence in some jurisdictions. Like their physical counterparts, digital signatures are indeed used to authenticate the sending of a message, but their strength as a primitive protocol does not end here. Since their introduction in 1976 by Whitfield Diffie and Martin Hellman,[2] further applications have been found in the building of secure distribution schemes, digitally processed financial transactions, cryptocurrencies[3] and more. It is known that a primitive analogue to digital signatures was developed decades before Diffie and Hellman made their public contributions, with the earliest known notion of authentication by some form of digital signature being a challenge-response mechanism used by the US Air Force to identify friendly aircraft, as far back as 1952[4]. Remarkably, even national identity and national government systems can be built with digital signatures at their core, as witnessed in Estonia's use of blockchain-style security for their Identity Card system, and eResidency scheme offered to International visitors and investors[5].

A. Digital Signatures and Security Formalised

In the following section we define digital signature schemes and their relevant security notions, as well as providing formal schematics of simple implementations of such schemes from the literature.

Definition 1 (Digital Signature Scheme). A digital signature scheme is a cryptographic protocol consisting of two distinct algorithms:

- A signing algorithm, in which the signing party (Alice), given a message (and typically a private key), produces a signature
- A verification algorithm, in which, given the message and signature, the verifier (Bob) either accepts or rejects Alice's claim of authenticity of the message

Digital signatures can fall into one of two categories, based on parties involved:

- True signatures: Requiring only two parties, Alice (the signer) and Bob (the receiver), true signature schemes involve the transmission of information directly from Alice to Bob, typically in the form of a message-signature pair and most often using asymmetric key cryptosystems (public key cryptography)
- Arbitrated signatures Requiring a trusted third party, Charlie (the arbiter), this type of scheme involves two distinct rounds of communications: Alice's communication to Charlie, and Charlie's communication to Bob. In this setup, Charlie provides verification to Bob, and the landscape is opened up for the use of symmetric key cryptosystems (private key cryptography).

Digital signature schemes are typically preceded by some form of key generation (and distribution if necessary), allowing us to express all signature schemes in terms of the following three steps:

- GEN: A key generation algorithm producing a private key (or set of private keys) and, if necessary, public keys.

- SIGN: Signature generated with a signing algorithm, and sent to Bob.
- VER: Bob receives the signature, and follows a verification algorithm before deciding whether or not to trust Alice's claim.

For any signature scheme to be considered secure and trustworthy for use, we require the scheme to provide the following under any and all conditions:

1. Authenticity: The receiver, Bob, when accepting a signature from Alice is convinced that the author of the message was indeed Alice.
2. Integrity: The receiver, Bob, can have faith that the message has not been altered since it left Alice.
3. Non-repudiation: Once a genuine signed message has left Alice, she has no way to convince Bob that she was in-fact not the author.

One more property often sought in signature schemes, but not strictly required for security, is for the signature to be able to be transferred. A signature scheme satisfying the above three conditions will convince Bob that Alice is indeed the author of the message, but transferability provides the ability for Bob to convince a third party, Charlie, that the message is indeed from Alice, without compromising the security of the system.
Attacks on signature schemes are known to typically fall into one of the four following categories:

1. Key-only attack: An adversary, Eve, knows only the public key of Alice (the signer)
2. Known-signature attack: Eve has access to Alice's public key, and message-signature pairs produced by Alice.
3. Chosen-message attack: Eve may choose a list of messages $(m_1, m_2, ..., m_l)$, which she will ask Alice to sign.
4. Adaptively-chosen-message attack: Similar to the above, except Eve has knowledge to adaptively choose messages based on the resulting message-signature pair of the previously signed message, allowing her to perform cryptanalysis with greater precision.

And we may describe the level of success achieved by Eve, from greatest success to least success, as follows:

1. Secret key knowledge: Eve discovers all of the secret information (typically Alice's secret key).
2. Universal forgery: Eve is able to forge the signature of any message, but lacks the secret key itself.
3. Selective forgeries: Eve can forge the signature for some messages of her choosing, but cannot do this arbitrarily.
4. Existential forgery: Eve may forge the signature for at least one message, but lacks the ability to choose this message from the set of all possible messages.
5. Failure: Eve finds out nothing about the secret information, and fails to forge a convincing signature for any message.

Clearly, Eve achieving universal forgery or above would render the signature scheme completely invalid, as she could go on to convince Bob (and other parties) that Alice has signed any message (or, at least fail to be rejected with utmost confidence). When discussing full security for a signature scheme, it is typical to demand it not allow any form of success, i.e., not even existential forgery, under any computing assumptions (or none).

That existential forgery is considered not permissible may seem a somewhat "strong" requirement; we could easily suppose that, given Eve's inability to choose a message, we could simply require a very large message space and propose that a message containing "gibberish" would not be accepted by Bob. In the case of sending email communications, this may, at first, seem suitable. It seems reasonable for Bob to expect Alice's message to make sense in their chosen language, and given Eve has no control over the message contents, we might expect her to have difficulty randomly selecting a perfectly coherent message. However, given a scenario in which Alice is simply sending a number, related to an amount in currency she is requesting Bob send her, an existential forgery would carry great threat! Eve might not be able to choose a precise amount, but it would be hard for Bob to label a string of integers as nonsensical.

B. Asymmetric Cryptography and Digital Signatures

With research in digital signatures growing alongside research in public key cryptography[4], the majority of well-known and well-studied signature schemes arise from public key cryptosystems. These typically rely upon certain mathematical assumptions about the hardness of problems (signature schemes based on symmetric encryption are generally reserved for arbitrated set-ups). An often-seen method of building a signature scheme is as follows: Find some public key cryptosystem based on one-way functions or trap-door functions, generate a signature using Alice's private key in the system, and allow any party to verify that Alice indeed sent the message using the publicly-shared encryption key. Well-known cryptosystems used in such a way include RSA[6], ElGamal[7], Rabin[8], and Fiat-Shamir[9]. We remark that (in a simplified manner), the main property that distinguishes a one-way function from a trap-door function is the existence of trap-door knowledge, some secret that allows the (usually) hard to invert function to become easily invertible.

1. Modular Exponentiation and the RSA Cryptosystem

A variety of trapdoor functions can be built based on performing exponentiation modulo n, depending on how we choose n. The simple act of squaring modulo n where $n = pq$ for some prime p, q forms a trap-door function, in which the trap-door knowledge is the prime factors (p and q). We can build further trap-door functions with different exponents by carefully choosing the exponent. Again, working in modulo n such that $n = pq$ for large primes p, q, if we choose some e such that e is coprime with $\phi(n) = (p-1)(q-1)$ (the Euler totient function of n) we find that for any given x,

$$c = x^e \pmod{n}$$

is a trapdoor function, where the trapdoor is once again the prime factors p, q. This forms the basis of the RSA cryptosystem. Whilst the work of Diffie and Hellman in 1976 may have built the theoretical bench on which research could seek to implement digital signatures, it was a later paper by Rivest, Shamir and Adleman that first exemplified a proof-of-concept on top of this workbench. The well-celebrated RSA paper[6] published in 1978 marked an early showcasing of asymmetric cryptosystems, well establishing the idea of public-key cryptography in a format that is in widespread use today. Relying on exponentiation under some $n = pq$ for large primes p and q, the RSA cryptosystem can be used to send encrypted data securely (under assumptions), and the same methodology can be used to implement a signing algorithm securely. For the basis of an RSA-Implemented cryptosystem, private keys and public keys must be created for use in the trap-door function, all of which is formalised as follows:

- Trap-door function: In the case of RSA, we take the trap-door function on some bit-string m to be

  $$\mathrm{RSA}_{\{E,D\}}(m) = m^{e,d} \pmod{n}$$

  Call RSA_E the RSA encryption function using key e, and RSA_D the RSA decryption function using key d, defined below. We require that $n = pq$ for some large primes p and q, and choose e such that for $\phi(n) = \phi(pq) = (p-1)(q-1)$, we have $\gcd(e, \phi(n)) = 1$, where gcd means greatest common divisor and $1 < e < \phi(n)$.

- Public Key: RSA takes as its encryption key the above chosen value, e. e is part of the public information in the cryptosystem, along with our chosen n for modular arithmetic.

- Decryption key: One calculates the decryption key, d, by determining the multiplicative inverse of e mod $\phi(n)$, i.e., determining $d \equiv e^{-1} (\mathrm{mod}\ \phi(n))$. The decryption exponent, along with the prime factors p, q, of n, are kept secret. d can be easily calculated when p and q are known.

We denote a message as m, with C being the resulting ciphertext following encryption on m, and S a signature (generated from a message m). The leading motivation for the RSA cryptosystem is its use for easy encryption of a message. Anyone with knowledge of the publicly shared information (e and n) can easily encrypt a message m by performing $C = m^e \mod n$. The intended recipient of the secret message, Alice, can easily recover $m = C^d \mod n$, and anyone lacking knowledge of d who intercepts C will struggle to find m from just the public knowledge. Loosely, the "hardness" of recovering m without d relies upon the hardness of discovering d without knowledge of p and q. We then see that, really, the security of the RSA cryptosystem reduces down to the intractability of factoring n into its prime factors p, q. This form of problem reduction is seen throughout digital signatures, and indeed all of cryptography.

Whilst the above demonstrates RSA's use as a tool to allow any party to transmit secret messages to a recipient Alice, it is easy to use the same tools to allow Alice to sign a message that can be verified by any party. Suppose Alice seeks to send a message, m, to some party Bob who wishes to verify that this message was not sent by some third party. This can be achieved by both parties carrying out the following:

1. Utilising her secret decryption key, Alice can now compute $\mathrm{RSA}_D(m) = m^d \mod n = S_m$.

2. Alice sends the message m to Bob, along with the associated signature, S_m.

3. Bob simply calculates $\mathrm{RSA}_e(S_x) = S_x^e \mod n = (m^d)^e \mod n = m^{d \cdot e} \mod n = m$

From this, it is clear Bob can be convinced that only Alice (or someone with Alice's secret decryption key d) could have sent this message. As e and n are public knowledge, any other party may also be convinced of this, allowing transferability of the signature.

This (simplified) view of RSA demonstrates many fundamentals of digital signatures within a classical framework: The need for a cryptosystem whose security we have good reason to believe in, even if it is not provable (the assumption of intractability presents issues for provable security, see II F), the ability to use this cryptosystem for signing (or at least modifying the cryptosystem for signing) and the need to ensure the three pillars of security for digital signatures: authenticity, integrity, and non-repudiation, whilst also hoping for the (at times less essential) property of transferability.

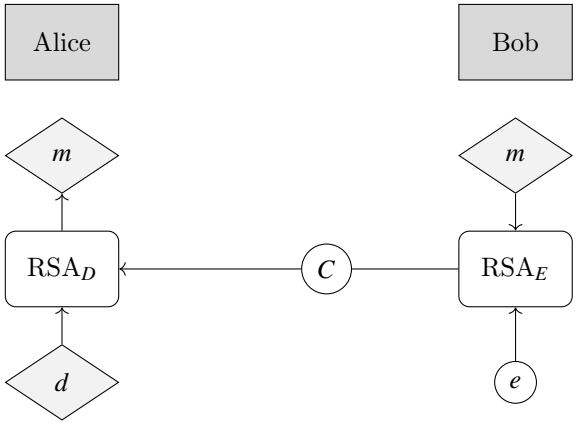

Figure 1: Schematic demonstrating communications secured using the RSA protocol. Bob encrypts (E) a (private) message m, using the RSA_E encryption function and the public key, e. The encrypted message, C, can now be sent publicly to Alice, who uses the RSA_D decryption function and the private key d, to retrieve m. Circles represent information that can be presented publicly, whilst diamonds must remain private.

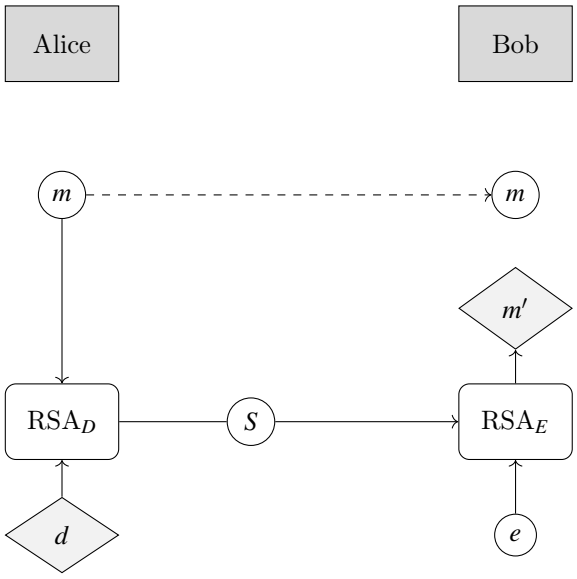

Figure 2: Schematic demonstrating message (m) signing using the RSA protocol. Alice computes a signature S using the RSA_D decryption function, her private key d and a (public) message m. Both m and S can now be sent to Bob via a public channel. Bob can now compute m' to be stored privately, using the RSA_E encryption function, the retrieved signature S and the public key e. If m' matches m closely (according to some pre-determined error-rate), the signature is accepted as valid. Otherwise, Alice is not accepted as the author of m. Circles represent information that can be presented publicly, whilst diamonds must remain private. Note the public and private variants of the message on Bob's end of communications. The private m' is calculated from the signature, and its value is checked against the publicly sent m.

2. *A note on symmetric digital signatures*

Whilst much of this section, along with the literature, focuses on asymmetric signature schemes whose roots lie in public key cryptosystems, this does not mean symmetric signature schemes arising from private key cryptosystems have no value in both research and application. Given the context (Alice signing a message, Bob verifying) it is easy to see why a private-key cryptosystem utilising the same (symmetric) key for both encryption and decryption is considered a weak arrangement for digital signatures: That both Alice and Bob use the same encryption key means either party can imitate the other. As long as Bob knows the encryption transformation used by Alice, he can always use the private key to generate a signature to deceive an unwitting third party, Charlie. The sought-after property of transferability is clearly lost. The potential applications for (secure) symmetric signatures is much smaller than that of asymmetric signatures. Both parties must trust each other to not be deceitful, which is far from practical for most settings (especially given the parties may have no knowledge of each other prior to the signing and verification). However, such systems are in use: for financial institutions they can be very beneficial as they are (often) less computationally taxing than their public key counterparts, and given a scenario where neither party has any reason to doubt the other's intentions (such as an ATM communicating with its parent financial institution), they can be used to great effect.

C. Other Bases of Schemes

1. *Modular Squaring*

The function $x \to x^2 \mod n$ for some $x \in \mathbb{Z}$ and $n = pq$, given p, q are prime, forms a trap-door function in which the trap-door information, like in RSA, is knowledge of the prime factors p, q of n. Rabin[8] introduced a cryptosystem whose core is reliant on utilising squaring under modular arithmetic as a trap-door function. Use of Rabin's system, along with some hashing function, can produce a signature scheme with certain advantages over RSA. It is unknown whether or not breaking RSA is actually as difficult as factoring: the security reduction that reduces RSA to factoring remains unproven as the RSA assumption. The potential that there may exist a security reduction from RSA to an easier problem than factoring is of concern. However, Rabin proved that breaking his cryptosystem is as difficult as the factoring problem. Thus, unlike RSA, cryptographers can find strength in its security as long as factoring remains intractible.

2. The Discrete Logarithm Problem

Throughout this section, we have treated the RSA cryptosystem as a tool by which to lay out the general case for, and elucidate on general points within, digital signature schemes. Principally, we have opted for this approach as the RSA cryptosystem is a well-studied example of a cryptosystem built upon the notion of a trapdoor function. As we have previously covered, this constitutes a popular method of construction, allowing us to perform both encryption and signing. Yet not all public-key cryptosystems, or all cryptosystems used to deploy signature schemes, must be reliant upon trapdoor functions. The discrete logarithm problem is one such example of a one-way function with no trapdoor information that is successfully implemented in widely-used signature schemes. Working within finite fields, and letting p be a prime and g some primitive root in $\mathbb{Z}_p^*$, the function

$$\mathrm{dExp} : \mathbb{Z}_{p-1} \to \mathbb{Z}_p^*, x \mapsto g^x$$

is a one-way function with no trap-door knowledge. dExp (the discrete exponential function) is easy to compute under a finite field, but its inverse, dLog, is believed to be intractable (the discrete analogue of the logarithmic function is hard to compute, i.e., it is hard to find x from $\mathrm{dExp}(x) = g^x$) and there exists no "trap-door" information that makes this inverse easily computable. The assumption about the intractability of computing x is known as the discrete logarithm assumption, and can form the basis of a cryptosystem, most notably one devised by ElGamal[7], and has applications in signing of information.

D. Security and Attacks across Asymmetric Signing

1. Signature Scheme Resistant to Adaptive Chosen Message Attacks

Adaptively chosen-message attacks, as defined in II A, utilise cryptanalysis of message-signature pairs (signed by Alice) as a powerful tool in a malicious party's (Eve's) pursuit of existential forgery against a given digital signature scheme. It is known that many well-studied cryptosystems are susceptible to such an attack, including RSA. In 1988 Goldwasser, Micali and Rivest (henceforth GMR) gave a thorough treatment of their signature scheme[10], whilst proving its security against adaptive chosen message attacks. Like many of the schemes preceding them, GMR's is reliant on trapdoors. However, GMR introduced the notion of claw-free permutation pairs[11], and claimed that signature schemes utilising trap-doors and claw-free permutations could produce an additional degree of security against adaptively chosen-message attacks when compared to the then-traditional method of 'simple' trap-door schemes. Whilst the scheme itself isn't as simple and easily presentable as RSA, the basic notion behind the claw-free permutations is simple to see:

Definition 2. Given a set of numbers, (x, y, z), we call them a claw of two permutations f_0 and f_1 if

$$f_0(x) = f_1(y) = z.$$

Further, we define a pair of permutations f_0, f_1 to be claw-free if there exists no efficient algorithm for computing a claw given two permutations.

GMR proved that the existence of such permutations implies the existence of a signature scheme ε-secure against adaptively-chosen-message attacks, i.e., Eve achieves existential forgery with probability $< \varepsilon$. Additionally, they presented a method of construction for practical claw-free permutations, utilising mathematical theory relevant to quadratic residues (an extensively studied tool in number theory, cryptosystems and cryptanalysis[12]) in order to find piecewise functions

$$f_0(x) = g_0(x)x^2 \pmod{n}$$

and

$$f_1(x) = g_1(x)x^2 \pmod{n}$$

where g_0, g_1 are piecewise constant functions. There exist functions in the form of f_0, f_1 that form claw-free permutations[13]. GMR show, via contradiction, that Eve's attempts of cryptanlaysis to achieve existential forgery can be reduced to finding a claw for the pair of permutations, and thus fail, even if the trapdoor functions used independently remain vulnerable to adaptively-chosen-message attacks.[14]

2. Hashing

Typically when performing RSA with the RSA-encryption and decryption functions $RSA_{\{E,D\}}(m) = m^{\{d,e\}} \bmod n$ with message m, encryption and decryption exponents e, d respectively, and modulus n, we take n to be some 1024-bit number. We bear in mind that, if sending a message in some text-based language, we are left with (at most) 128 ASCII characters. Assuming the language chosen is well-defined with a set of rules, we can assume most documents that need signing will be greater than this stringent limit. In order to allow the signing of messages and documents of arbitrary length, cryptographers typically turn to hash functions.

Definition 3 (Hash). Simply, a Hash function, H, is a function taking in as its input some data of arbitrary length, and outputting a hash digest (or, simply, hash or digest) of a fixed length.

For use in cryptography, we generally seek the following three properties from a hash function:

- Pre-image resistance: Given a hash digest h, finding any message m with $h = H(m)$ should be a difficult task. (We can consider the similarity between this property, and that of the one-way function.)

- Collision resistance: The essence behind collision resistance is that there should be a very low probability of finding two messages outputting the same digest. Collision resistance is typically categorised into one of two groups: Weak collision resistance, in which for given a message m_1, it should be difficult to find a message m_2 with $H(m_1) = H(m_2)$ when $m_1 \neq m_2$; and Strong collision resistance, in which it should be difficult to find two messages $m_1 \neq m_2$ such that $H(m_1) = H(m_2)$.

Generally, it is favourable that these properties define a platform upon which a malicious adversary cannot modify the input data without changing the digest. Further, we desire a good distribution of digests, that is, given two n-bit-strings m_1 and m_2 with a small Hamming distance ε, we seek very different outputs, i.e. a (relatively higher) Hamming distance between $H(m_1)$ and $H(m_2)$. Clearly, the overarching goal of creating a good hash function is that an adversary should find it very hard to determine the input of a hash, and cryptanalysis by attacks involving similar messages should be unable to find a weakness here.

Full security in the random oracle model can be achieved using a full domain hash function, in which the image of the hash function is equal to the domain of the RSA function. However, most types of RSA widely used do not implement full-domain hash functions, instead opting for hash functions such as SHA, MD5, and RIPEMD[15–17].

3. Probabilistic Signatures

In 1996, Bellare and Rogaway introduced the notion of the probabilistic signature scheme (PSS)[18], in which the signature generated is dependent upon the message and a randomly chosen input. This results in a signature scheme whose output for a given message does not remain consistent over multiple implementations. Utilising a trap-door function (typically one well-used in non-probabilistic schemes, such as the RSA function), a hash function and some element of randomness (typically a pseudo-random bit generator), a signature scheme that is probabilistic in nature may be implemented. Such schemes can be used to sign messages of arbitrary length, and to ensure that the message m is not recoverable from just the signature of m. RSA-PSS is a common probabilistic interpretation of the RSA signing scheme that forms part of the PKCS standards published by RSA laboratories[19].

E. Cryptographic Standards and Modern Use

We have already discussed how hashing may be used before signing a message (along with padding) to ensure all messages signed are of an appropriate size. However, the use of hashing in digital signatures extends beyond the "Hash and sign" idea used for signing protocols such as RSA. A protocol introduced by Fiat and Shamir[9] has led to the creation of the Fiat-Shamir paradigm. The Fiat-Shamir paradigm takes an interactive proof-of-knowledge protocol[20] and replaces interactive steps with some random oracle, typically a publicly-known collision hash function. A thorough treatment of the paradigm can be found in Delfs' and Knebl's textbook on cryptography[21]. In addition to their use in creating signature schemes that are secure against adaptively-chosen-message attacks, it has been shown by Damgard[13] that claw-free permutations can play a role in creating collision-resistant-hash-functions (this should not seem too surprising, as it is easily recognised that their definitions are similar: Collision - resistant hashing can almost be seen as a single-function analogue of claw-free permutations).

The work of ElGamal on cryptosystems making use of the one-way nature of the discrete logarithm forms the basis of the Digital Signature Algorithm[22], a cryptographic standard popular since its proposal as a NIST submission for the Digital Signature Standard, DSS.

Recent years have seen an increased interest in electronic voting, a concept heavily reliant on signature schemes. Electronic voting typically requires a cryptosystem that is both probabilistic and holds homomorphic properties. ElGamal is a good example of an applicable cryptosystem. Electronic voting has been used in a variety of countries, including the US(the 2000 Democratic Primary election in Arizona[23] is often cited as a landmark event in internet voting); Scottish Parliament and local elections since 2007 (although the 2007 elections can be considered good proof as to why great care must go into researching the implementation of these systems before use[24]), Brazil[25] (whose 2010 presidential election results were announced just 75 minutes after polls closed thanks to electronic voting), and India, with the state of Gujarat being the first Indian state to enable online voting in 2011[26]. In Europe, Estonia also utilise electronic voting[5], with the idea of the Estonian digital ID-card, which provides a digital signature, being pivotal in how government and society are run in the Baltic country.

Another subfield of cryptographic research that has garnered increased interest in recent times is Elliptic Curve Cryptography. Schemes based on the discrete logarithm problem (such as ElGamal/DSA) can be implemented similarly on the mathematical framework

of elliptic curves[27] instead of finite fields. A key benefit of deploying a cryptosystem in such a way is the ability to perform computations at shorter binary lengths than traditionally used, without conceding security. This makes such schemes good candidates for when resources are limited, and Elliptic Curve DSA (ECDSA)[28] is an example of such a scheme that forms a cryptographic standard, and is included in the Transport Layer Security (TLS) protocol.[29]

In 1979, Ralph Merkle patented the concept of the hash tree, commonly known as the Merkle tree after him. Merkle trees can be paired with one-time signature schemes (within a symmetric cryptographic framework) to form a Merkle-Tree Based Signature scheme[30]. Such schemes still remain only suitable for one-time use, although the work of Naor and Yung explores an extension of these types of schemes to complete multi-use signature schemes. It is believed that such signature schemes may be resistant to quantum-attacks, which are mentioned below and discussed further in section 2.

F. Looking forward for security

As we have seen, the vast majority of widely implemented cryptographic algorithms (especially those that rise from public-key cryptosystems) rely upon unproven mathematical assumptions about the hardness of certain problems in order to provide us with security. This review is by no means expansive on the workings of different signature schemes under varied cryptosystems, and a reader seeking a thorough treatise of the field may turn to Simmons et al.[31] (for an exploration of early public key cryptosystems and signatures) and Delfs-Knebl[21] (for a treatise of modern cryptography, with extensive sections on signatures). With the increase in research in applications of quantum theory to modern technology, these previously held assumptions are left to fall apart in front of us. Since Deutsch's introduction of the Universal Quantum Computer,[32] research in utilising the power of quantum theory for computing has yielded many strong theoretical results, with early work including the development of algorithms for a quantum computer that can perform certain tasks faster than a classical computer is believed to be able to. Included in these is Shor's algorithm,[33] which can perform prime factorisation at a speed that would allow currently implemented cryptosystems to be broken. Whilst the practical implementation of such algorithms is yet to yield results strong enough to cause immediate worry, research is still looking forward to ensure security shall not be compromised as quantum computers grow more powerful.

III. POST-QUANTUM DIGITAL SIGNATURES

A. Introduction: A Problem

As previously mentioned, advances in quantum computing have raised concerns for the field of classical cryptography. Here we give a brief overview of why, followed by a discussion of the responses from the cryptographic community.

1. *Quantum Cryptanalysis of Classical Cryptography*

In an era where popular thinking was that problems based on factoring would be unbreakable, the introduction of Shor's algorithm[33] in 1994 caused uncertainty in the security of cryptosystems that were previously assumed to be secure. This review gives a brief overview of the techniques used.

Factoring a composite number N can be reduced to the problem of finding the period of a function. This is done by picking a random number $a < N$, checking that $\gcd(a,N) = 1$ (if $\gcd(a,N) \neq 1$ then we've found a factor of N and we're done), then looking for the period of the function:

$$f(x) = a^x \mod N. \tag{4}$$

Up to this stage this can all be achieved classically. The quantum Fourier transform is used to find the period, resulting in Shor's algorithm being extremely efficient and appearing in the complexity class BQP[34]. This is almost exponentially faster than the fastest known classical factoring algorithm, the general number field sieve[35].

This period solving algorithm can also be used to solve the discrete logarithm problem[36], which also breaks the hardness assumption of this problem. From this, Shor's algorithm can be extended to a more general problem: the Hidden Subgroup Problem (HSP)[37–39]. The HSP states that given a group G, a finite set X and a function $f : G \to X$ that hides a subgroup $H \leq G$, determine a generating set for H only given evaluations of f. We say that a function $f : G \to X$ hides H if, for all $g_1, g_2 \in G$, $f(g_1) = f(g_2)$ if and only if $g_1 H = g_2 H$. Within this framework, Shor's algorithm can be seen as solving the HSP for finite abelian groups. Other problems can similarly be generalised to this framework. For instance, if a quantum algorithm could solve the HSP for the symmetric group then one of the key hard problems for Lattice Cryptography (see section III C) - the shortest vector problem - would be broken[40].

Whilst Shor's algorithm has received the most attention for the problems it causes in cryptography, it is by far not the only quantum algorithm to attack current schemes. Grover's search algorithm[41,42] can be

used in certain schemes, and other factorisation algorithms such as the quantum elliptic-curve factorisation method[43] have had some success. We point the reader in the direction of Bernstein et. al.[44] and Jordan et. al.[45] for more complete surveys on quantum cryptanalysis of classical cryptography.

2. What's being done?

Whilst current estimates place the development of practical quantum computers capable of posing a security threat many years in the future (at time of writing the record for using Shor's algorithm to factor a 'large number' into two constituent primes stands at $21 = 3 \cdot 7$ [46], though much larger factorisations have been achieved in the adiabatic case[47]), it is pertinent to replace our current systems well in advance of that. In light of that, the National Institute for Standards and Technology (NIST) put out a call for submissions in 2016[48?] to attempt to set a new quantum-secure standard. This ongoing project aims to find new standards for both public key encryption and digital signatures.

The NIST evaluation criteria[49] for these new schemes sets out both required security levels and computational cost. For the security levels, it is assumed that the attacker has access to signatures for no more than 2^{64} chosen messages using a classical oracle. The security levels are grouped into broad categories defined by easy-to-analyse reference primitives - in this case, the Secure-Hash-2 (SHA2)[15] and the Advanced Encryption Standard (AES)[50]. The rationale behind the seemingly vague categories being that it is hard to predict advances in quantum computing and quantum algorithms, and so rather than using precise estimates of the number of 'bits of security', a comparison will suffice. See table I for the exact security levels.

Level	Reference Primitive	Security Equivalence
1	AES 128	Exhaustive key search
2	SHA 256	Collision search
3	AES 192	Exhaustive key search
4	SHA 384	Collision search
5	AES 256	Exhaustive key search

Table I: NIST security levels.

For quantum attacks, restrictions on circuit depth are given, motivated by the difficulty of running extremely long serial quantum computations. Proposed schemes are also judged on the size of the public keys and signatures they produce as well as the computational efficiency of the key generation.

As of July 22, 2020 the NIST project to set a new quantum-secure cryptographic standard entered the third round of submissions[51], with only six proposals for digital signatures remaining. These are further split into three finalists and three alternative candidates. The finalists are the algorithms which have shown the most promise and are general-purpose for the most part. Those in the alternative candidate track are considered either tailored to more specific applications or require more time to mature. Round three is predicted to take around eighteen months, though due to the ongoing global pandemic as of November 10, 2020 the schedule is much looser. Following the third round NIST aim to continue the review process, allowing for some alternative candidates to be standardised at later date as well as giving considerations to ideas that were developed too recently to be included in the initial round in 2016.

From these, two front runners have emerged for the mathematical basis which will replace our current systems: Multivariate and Lattice cryptography. See table II for how these fall into the different categories.

Track	Multivariate	Lattice	Other
Finalist	Rainbow[52]	Dilithium[53] FALCON[54]	
Alternative	MQDSS[55]		Picnic[56] SPHINCS$^+$[57]

Table II: NIST digital signature submissions by underlying mathematical structure type.

Here we will give a brief overview of both Multivariate and Lattice based cryptography, followed by a note on the other schemes.

B. Multivariate Cryptography

Multivariate Cryptography was developed in the late 1980's with the work of Matsumoto and Imai[58]. Originally named C^* cryptography after the first protocol, the name Multivariate Cryptography was adopted when the work of Patarin broke and then generalised the C^* protocol[59]. After Shor's now infamous algorithm was developed it was realised that the structure of Multivariate Cryptography could be used as a direct response. For a more in depth treatment of the subject, the reader may consult Bernstein et. al.[44] or Wolf[60].

All of the schemes are based on a hard problem which is relatively straightforward to understand (though several of the constructions extend the basic problem to more complex settings): the problem of solving multivariate quadratic equations over a finite field. That is, given a system of m polynomials in n variables:

$$\mathsf{P} = \begin{cases} y_1 = p_1(x_1, \ldots, x_n) \\ y_2 = p_2(x_1, \ldots, x_n) \\ \vdots \\ y_m = p_m(x_1, \ldots, x_n) \end{cases}, \tag{5}$$

and the vector $\mathbf{y} = (y_1, \ldots, y_m) \in \mathbb{F}^m$, find a solution $x \in \mathbb{F}$ which satisfies the equations above. Formally we say that for a finite field $\mathbb{F}$ of size $q := |\mathbb{F}|$, an instance of an $\mathsf{MQ}(\mathbb{F}^n, \mathbb{F}^m)$-problem is a system of polynomial equations of the form:

$$p_i(x_1, \ldots, x_n) = \sum_{1 \le j \le k \le n} \gamma_{ijk} x_j x_k + \sum_{j=1}^{n} \beta_{ij} x_j + \alpha_i, \quad (6)$$

where $1 \le i \le m$ and $\gamma_{ijk}, \beta_{ij}, \alpha_i \in \mathbb{F}$. These are collected in the polynomial vector $\mathsf{P} := (p_1, \ldots, p_m)$. Here, $\mathsf{MQ}(\mathbb{F}^n, \mathbb{F}^m)$ denotes a family of vectorial functions $\mathsf{P} : \mathbb{F}^n \to \mathbb{F}^m$ of degree 2 over $\mathbb{F}$:

$$\mathsf{MQ}(\mathbb{F}^n, \mathbb{F}^m) = \{\mathsf{P} = (p_1, \ldots, p_m) \mid \text{for } p \text{ of the form (6)}\} \quad (7)$$

Whilst theoretically the polynomials could be of any degree, there is a trade-off between security and efficiency. Higher degrees naturally have larger parameter spaces, but too low a degree would be too easy to solve and therefore would be deemed too insecure. Quadratics are chosen as a compromise between the two.

The final two pieces required for signatures are two affine maps, $S \in \mathrm{Aff}^{-1}(\mathbb{F}^n)$ and $T \in \mathrm{Aff}^{-1}(\mathbb{F}^m)$. Both of these can be represented in the usual way:

$$S(x) = M_S x + v_S \quad (8)$$

$$T(x) = M_T x + v_T \quad (9)$$

where $M_S \in \mathbb{F}^{n \times n}$, $M_T \in \mathbb{F}^{m \times m}$ are invertible matrices and $v_S \in \mathbb{F}^n$, $v_T \in \mathbb{F}^m$ are vectors.

For most multivariate signature schemes, the secret key is the triple $(S^{-1}, \mathsf{P}'^{-1}, T^{-1})$, where S and T are affine transforms and P' is a polynomial vector (defined similarly to equation (5)), known as the central equation. The choice of the shape of this equation is largely what distinguishes the different constructions in multivariate cryptography. The public key is then the following composition:

$$\mathsf{P} = S \circ \mathsf{P}' \circ T. \quad (10)$$

To forge the signature, one would have to solve the following problem: for a given $\mathsf{P} \in \mathsf{MQ}(\mathbb{F}^n, \mathbb{F}^m)$ and $\mathbf{r} \in \mathbb{F}^m$ find, if any, $\mathbf{s} \in \mathbb{F}^n$ such that $\mathsf{P}(\mathbf{r}) = \mathbf{s}$. It was shown in Lewis et. al.[61] that the decisional form of this problem is NP-hard, and it is believed to be intractable in the average case[62].

We now formally outline the general scheme for signatures based on the Multivariate Quadratic problem:

i) Alice generates a key pair $(\mathsf{s}_k, \mathsf{p}_k)$, where $\mathsf{s}_k = (S^{-1}, \mathsf{P}'^{-1}, T^{-1})$ and $\mathsf{p}_k = \mathsf{P} = S \circ \mathsf{P}' \circ T$, then distributes p_k.

ii) Alice then hashes the message, m, to some $c \in \mathbb{F}^n$ using a known hash function, then computes:

$$s = \mathsf{P}^{-1}(c) = T^{-1}(\mathsf{P}'^{-1}(S^{-1}(c))),$$

sending the pair (m, s) to Bob.

iii) Bob then needs to check that $\mathsf{P}(s) = c = H(m)$ for the known hash function H.

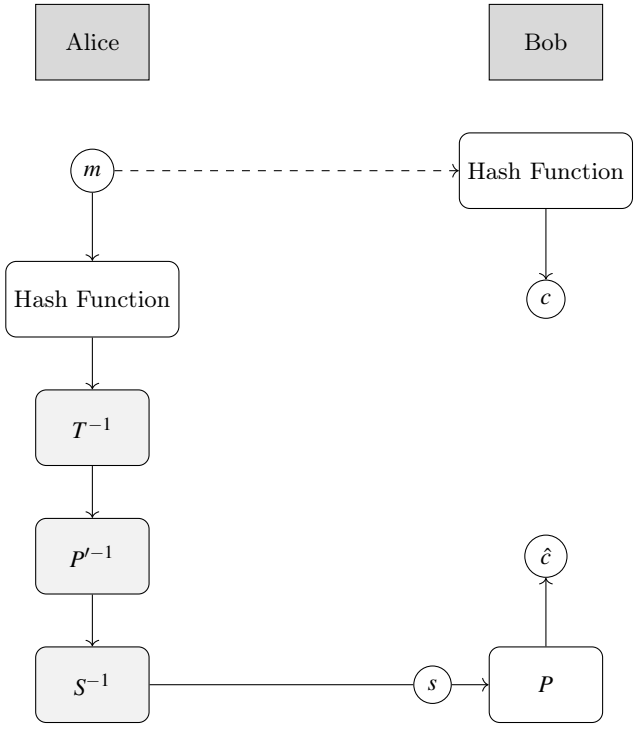

Figure 3: A general schematic for Multivariate Quadratic signature schemes. Alice hashes the message m to some vector $c \in \mathbb{F}^n$, which is transformed under the affine transform $c' = T^{-1}(c)$. The central equation is then applied, $s' = \mathsf{P}'(c')$, followed by a second affine transformation to create the signature $s = S^{-1}(s')$. To check the signature Bob only has to recover $\hat{c} = \mathsf{P}(s)[= S(\mathsf{P}'(T(s)))]$ and confirm that it matches the hash value $H(m) = c$.

See figure 3 for a diagram of how the signature schemes work. This forms the backbone of most multivariate schemes. The following subsections examine several of the adaptations of this framework employed in various NIST submissions.

1. *Unbalanced Oil and Vinegar*

The Oil and Vinegar scheme was first introduced by Patarin[63], but was broken by Kipnis and Shamir[64] and generalised to the now common Unbalanced Oil and Vinegar (UOV) protocol.

Definition 11 (Unbalanced Oil and Vinegar). Let $\mathbb{F}$ be a finite field, $o, v \in \mathbb{N}$ such that $o + v = n$ and $\alpha_i, \beta_{ij}, \gamma_{ijk} \in \mathbb{F}$ for $1 \le i \le v$ and $1 \le j \le k \le n$. Polynomials of the

following form are central equations in the UOV-shape:

$$p_i(x_1,\ldots,x_n) := \sum_{j=1}^{v}\sum_{k=1}^{n}\gamma_{ijk}x_jx_k + \sum_{j=1}^{n}\beta_{ij}x_j + \alpha_i. \qquad (12)$$

The first $x_1,\ldots,x_v$ terms are known as the vinegar terms and the second register of $o = n - v$ terms are called the oil terms. If $o \neq v$ it is called unbalanced.

In these equations, the vinegar terms are combined quadratically with themselves, and then combined quadratically with the oil terms, whereas the oil terms are never mixed with themselves. For a secure construction, the required discrepancy between the number of oil and vinegar terms is $v \geq 2o$. Unbalanced Oil and Vinegar has become one of the most common constructions for Multivariate Cryptography, and it has itself become a way of varying other constructions by putting them in an UOV-shape. The NIST submission Rainbow is largely based on the unblanced oil and vinegar scheme. One of the major problems with UOV is the length of the signatures and the key sizes, and both of these submissions get around this by introducing additional structure on top of the UOV shape. For a comparison of signature and key size as well as other efficiency markers, see section III F.

2. Hidden Field Equations

The Hidden Field Equations (HFE) protocol[59] is a generalisation of one of the original multivariate systems, the Matsumoto-Imai scheme[58]. Similar to Oil and Vinegar, the underlying scheme was broken before the underlying trapdoor was generalised. However, unlike UOV, this scheme uses more than one field: the ground field $\mathbb{F}$ and it's n^{th}-degree field extension $\mathbb{E}$, that is $\mathbb{E} := \mathbb{F}[t]/f(t)$ where $f(t)$ is an irreducible polynomial over $\mathbb{F}$ of degree n.

Definition 13 (Hidden Field Equations (HFE)). Let $\mathbb{F}$ be a finite field with $q := |\mathbb{F}|$ elements, $\mathbb{E}$ its n^{th}-degree extension field and $\phi : \mathbb{E} \to \mathbb{F}^n$ the canonical, coordinate-wise bijection between the extension field and the vector space. Let $P(X)$ be a univariate polynomial over $\mathbb{E}$ with:

$$P'(X) := \sum_{\substack{0\leq i,j\leq d \\ q^i+q^j\leq d}} C_{ij}X^{q^i+q^j} + \sum_{\substack{0\leq k\leq d \\ q^k\leq d}} B_kX^{q^k} + A, \qquad (14)$$

where

$$\begin{aligned} C_{ij}X^{q^i+q^j} &\quad \text{for } C_{ij} \in \mathbb{E} \text{ are quadratic terms,} \\ B_kX^k &\quad \text{for } B_k \in \mathbb{E} \text{ are linear terms, and} \\ A &\quad \text{for } A \in \mathbb{E} \text{ is constant,} \end{aligned}$$

for $i, j \in \mathbb{N}$ and a degree $d \in \mathbb{N}$. We say that central equations of the form $\mathsf{P}' := \phi \circ P' \circ \phi^{-1}$ are in HFE shape.

The GeMSS submission to the NIST proceedings uses Hidden Field Equations, although it adapts the form using 'minus and vinegar modifiers'[65]. This has allowed the design to become more flexible in its choice of security parameters whilst improving efficiency.

3. Attacks

The cryptanalysis of multivariate schemes comes in two forms:

a. Structural These focus on taking advantage of the specific structural faults in the design on different protocols. Included amongst this are attacks on a form of Multivariate cryptography called MINRANK[66] and the hidden field equations[67].

b. General Attacks that directly try and break the underlying hardness assumption of solving multivariate equations. These include the use of techniques such as utilising Gröbner bases to make the solving of the multivariate systems easier. For a good overview of the area see Billet and Ding[68].

C. Lattice Cryptography

Lattice cryptography was first introduced by the work of Ajtai[69] who suggested that it would be possible to base the security of cryptographic systems on the hardness of well-studied lattice problems. The familiarity of these problems made them an attractive candidate for PQC. This led to the development of the first lattice-based public-key encryption scheme - NTRU[70]. However, this was shown to be insecure and it would take the work of Regev to establish the first scheme whose security was proven under worst-case hardness assumptions[71]. For an overview of the field of lattice cryptography, we direct the reader to Peikert[72].

There is a whole suite of lattice problems on which cryptographic schemes are based. Here - following some basic definitions - we will introduce the key ideas that form the foundation of contemporary Lattice Cryptography.

Definition 15. A lattice $\Lambda \subset \mathbb{R}^n$ is a discrete additive subgroup of $\mathbb{R}^n$. That is $\mathbf{0} \in \Lambda$, if $\mathbf{x}, \mathbf{y} \in \Lambda$ then $-\mathbf{x}, \mathbf{x}+\mathbf{y} \in \Lambda$, and any $\mathbf{x} \in \Lambda$ has a neighbourhood of $\mathbb{R}^n$ which has no other lattice points.

We note here that lattices can be more generally defined as a discrete additive subgroup of some general vector space V, but are most commonly restricted to $\mathbb{R}^n$. Any non-trivial lattice is countably infinite, however each lattice can be finitely generated by all the integer combinations of some set of vectors in $\mathbb{R}^n$, $\mathbf{B} = \{\mathbf{b}_1,\ldots,\mathbf{b}_k\}$ for some $k \leq n$. Typically $k = n$, in which case we call Λ a full rank lattice. We call $\mathbf{B}$ the basis for a lattice Λ and express it as a matrix of row vectors. Often we

describe the lattice as a linear sum of integer multiplied basis vectors, writing:

$$\Lambda(\mathbf{B}) = \left\{ \sum_{i=1}^{k} z_i \mathbf{b}_i \middle| z_i \in \mathbb{Z} \right\}. \tag{16}$$

A basis $\mathbf{B}$ isn't unique as any two bases $\mathbf{B}_1$, $\mathbf{B}_2$ for a lattice Λ are related by a unimodular matrix $\mathbf{U}$ such that $\mathbf{B}_1 = \mathbf{U}\mathbf{B}_2$. Another crucial lattice definition is the dual lattice:

Definition 17. Given a lattice Λ in V where V is endowed with some inner product $\langle \cdot, \cdot \rangle$, the dual lattice $\Lambda^\perp$ is defined as $\Lambda^\perp = \{\mathbf{v} \in V \mid \langle \Lambda, \mathbf{v} \rangle \subset \mathbb{Z}\}$.

Here we give a quick note about additional structure that can be imbued in lattices. A common technique is to construct lattices embedding algebraic structures, such as rings and modules. It is beyond the scope of this paper to go into detail on how one constructs these, so we point the readers in the direction of Lubashevksy et. al.[73] and Grover[74] for a more complete understanding of the structure. The justification for these will become clear once we introduce lattice hard problems.

Another important concept is that of Discrete Lattice Gaussians:

Definition 18 (Discrete Lattice Gaussian). Given a basis $\mathbf{B}$ for a lattice $\Lambda(\mathbf{B})$, mean $\mu \in \mathbb{R}^n$ and standard deviation $\sigma > 0$, the discrete Gaussian over a lattice is defined as,

$$D_{\Lambda,\sigma,\mu}(\mathbf{x}) := \frac{\rho_{\sigma,\mu}(\mathbf{B}\mathbf{x})}{\rho_{\sigma,\mu}(\Lambda)}, \quad \mathbf{x} \in \mathbb{Z}^n, \tag{19}$$

where $\rho_{\sigma,\mu}(\mathbf{y}) := \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y}-\mu\|\right)$ and $\rho_{\sigma,\mu}(\Lambda) = \sum_{\mathbf{x}\in\mathbb{Z}^n} \rho_{\sigma,\mu}(\mathbf{B}\mathbf{x})$.

Discrete lattice Gaussian sampling is one of the core features of Lattice Cryptography, being employed in some manner in most schemes. However, this form of sampling comes with a whole host of issues. For one, it is computationally hard to sample directly from such distributions, leading to algorithms that sample from statistically close distributions[75]. Unfortunately, these approximate distributions aren't necessarily spherical Gaussians and therefore have the potential to leak information about the secret[76]. Other inherent problems include finding the upper and lower bounds on the choice of variance - too low a variance also leaks information, but too high a variance will produce signatures that are insecure (see later sections). See Prest[77] for a more comprehensive discussion on discrete Gaussian sampling.

1. Fundamental Hard Problems

We now move on to the hard lattice problems. First, we give a few of the fundamental hard problems, which form a foundation for the hard problems that contemporary lattice cryptography is built on.

Definition 20 (Shortest Vector Problem (SVP)). Define $\lambda_1(\Lambda)$ to be the length shortest non-zero vector in Λ. The Shortest Vector Problem (SVP) is: given a basis $\mathbf{B}$ of a lattice Λ, compute some $\mathbf{v} \in \Lambda$ such that $\|\mathbf{v}\| = \lambda_1(\Lambda)$.

Here, $\lambda_m(\Lambda)$ are the succesive minima of the lattice, where each vector $\|v_i\| = \lambda_i \leq \lambda_j$ for $i < j$, with $\lambda_1(\Lambda)$ being the shortest vector in the lattice. The norm function $\|\cdot\|$ is left intentionally unspecified, though it is typically the Euclidean norm. This problem is regarded as hard in both a classical and quantum setting, but it falters when applied to cryptographic schemes with some probabilistic element. It is more common to use the approximate analogue to the SVP, which is as follows:

Definition 21 (Approximate Shortest Vector Problem (SVP_γ)). Given a basis $\mathbf{B}$ of a lattice $\Lambda(\mathbf{B})$, find a nonzero vector $\mathbf{v} \in \Lambda$ such that $\|\mathbf{v}\| \leq \gamma(n) \cdot \lambda_1(\Lambda)$.

We also make mention of bounded distance decoding (BDD) which asks the user to find the closest lattice vector to a prescribed target point $\mathbf{t} \in \Lambda$ which is promised to be 'rather close' to the lattice.

Definition 22 (Bounded Distance Decoding (BDD_γ)). Given a basis $\mathbf{B}$ of a full-rank lattice $\Lambda(\mathbf{B})$ and a target vector $\mathbf{t} \in \mathbb{R}^n$ with a guarantee that $\mathrm{dist}(\mathbf{t}, \Lambda) < d = \lambda_1(\Lambda)/(2\gamma(n))$, find unique lattice vector $\mathbf{v} \in \Lambda$ such that $\|\mathbf{t} - \mathbf{v}\| < d$.

The above problems have varying degrees of provable hardness. The exact version of the shortest vector problem is known to be NP-hard under randomised reductions[78], however, the implementation of the hard problems such as bounded distance decoding relies on polynomial encoding so it is in the complexity class $\mathsf{NP} \cap \mathsf{co}-\mathsf{NP}$[79]. Currently, there are no known quantum algorithms that solve any of the above problems in polynomial time, but there have been various attempts, see section III C 5 for further detail.

2. Foundations of Contemporary Lattice Crypto

Whilst the previous hard problems are fundamental to lattices, they are not easily implementable in lattice schemes. Here we introduce the two problems which form the foundation for contemporary lattice cryptography: the short integer solution (SIS)[69] and learning with errors (LWE)[71].

Definition 23 (Short Integer Solution ($\mathsf{SIS}_{n,q\beta,m}$)). Given m uniformally random vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$ forming the columns of a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, find a nonzero integer vector $\mathbf{z} \in \mathbb{Z}^m$ of norm $\|\mathbf{z}\| \leq \beta$ such that:

$$f_{\mathbf{A}}(\mathbf{z}) := \mathbf{A}\mathbf{z} = \sum_i \mathbf{a}_i \cdot z_i = \mathbf{0} \in \mathbb{Z}_q^n. \tag{24}$$

LWE is an average-case problem introduced by Regev, often referred to as the 'encryption enabling' analogue of the SIS problem.

Definition 25 (LWE distribution). For a vector $\mathbf{z} \in \mathbb{Z}_q^n$ called the secret, the LWE distribution $A_{\mathbf{z},\chi}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is sampled by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \longleftarrow \chi$, and outputting $(\mathbf{a}, b = \langle \mathbf{z}, \mathbf{a} \rangle + e \mod q)$.

The problem comes in two distinct forms: decision and search. Decision requires distinguishing between LWE samples and uniformly random ones, whereas search requires finding a secret given LWE samples.

Definition 26 (Decision $\mathsf{LWE}_{n,q,\chi,m}$). Given m independent samples $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ where every sample is distributed according to either:

i) $A_{\mathbf{z},\chi}$ for a uniformly random $\mathbf{z} \in \mathbb{Z}_q^n$ (fixed for all samples),

ii) The uniform distribution,

distinguish which is the case.

Definition 27 (Search $\mathsf{LWE}_{n,q,\chi,m}$). Given m independent samples $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ drawn from $A_{\mathbf{z},\chi}$ for a uniformly random $\mathbf{z} \in \mathbb{Z}_q^n$, find $\mathbf{z}$.

The learning with errors problem has been shown to be at least as hard as quantumly solving SVP on arbitrary n-dimensional lattices. The following security reduction can be shown following the proof from Regev[71]:

$$\begin{array}{ccccccc} \mathsf{SVP} & \longleftarrow & \mathrm{DGS} & \longleftarrow & \mathsf{BDD} & \longleftarrow & \text{Search LWE} \\ & & & & & & \uparrow \\ \text{Worst-Case Decision LWE} & & \longrightarrow & & & & \text{Decision LWE} \end{array}$$

Here DGS stands for discrete Gaussian sampling. We note that the reduction between discrete Gaussian sampling and the BDD problem is a quantum step.

As previously mentioned, it is possible to construct lattices from specific algebraic structures such as rings or modules[80], when combined with the above problems it is known as structured-LWE or structured-SIS. This is largely done for efficiency reasons as the parameter space needed to implement systems based on these structures is greatly reduced[73]. The choice of which structure is worth using has some nuances, however. For example, ring-LWE is generally considered to be more efficient than module-LWE, however the efficiency comes at a cost in flexibility and security. Increasing the security of a scheme requires increasing the dimension of the lattice, which in the ring case is often chosen such that $N = n$ where $n = 2^k$ for some integer k. Thus, going up a security level requires going from dimension 512 to 1024 for instance, whereas a more optimal scheme may lie inbetween these. Module-LWE has dimension parametrised by an integer d such that $N = dn$, again for n of the form 2^k. Setting $d = 3$ and $n = 256$ allows for a total dimension of 768, which may be preferable for the targeted level of security. Even further structure can be imbued which may yet give greater flexibility: middle-product-LWE[81] and cyclic-LWE[82]. The security of all of these schemes based on algebraic structure has been questioned however, as the reduction to standard LWE is not fully understood[83–85].

All of the lattice based NIST submissions for signature schemes use some kind of structure: qTESLA and FALCON are based on rings (FALCON, however, uses a distinction between binary and ternary forms to capture the intermediate security levels) whereas Dilithium is based on module-LWE.

3. The GPV Framework

Introduced in the seminal paper of Gentry et. al.[86], the Gentry-Peikert-Vaikuntanathan (GPV) framework gives an overarching structure for taking advantage of 'natural' trapdoors in lattices to obtain signatures. It is built on a signature scheme first introduced in Goldreich-Goldwasser-Halevi (GGH)[87] and NTRUsign[88] schemes:

- The public key is a full rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ generating a lattice Λ. The private key is a matrix $\mathbf{B} \in \mathbb{Z}_q^{m \times m}$ generating $\Lambda_q^\perp$, the dual lattice of $\Lambda \mod q$.
- Given a message m, a signature is a short value $\mathbf{s} \in \mathbb{Z}_q^m$ such that $\mathbf{s}\mathbf{A}^T = H(m) = \mathbf{c}$ where $H : \{0,1\}^* \to \mathbb{Z}_q^n$ is a known hash function. Given $\mathbf{A}$, verifying $\mathbf{s}$ as a valid signature is straightforward: check $\mathbf{s}$ is short and that $\mathbf{s}\mathbf{A}^T = \mathbf{c}$.
- Computing a signature requires more care however:
  i) Compute an arbitrary preimage $\mathbf{c}_0 \in \mathbb{Z}_q^m$ such that $\mathbf{c}_0\mathbf{A}^T = \mathbf{c}$. $\mathbf{c}_0$ is not required to be short so it can be computed with relative ease.
  ii) Use $\mathbf{B}$ to compute a vector $\mathbf{v} \in \Lambda_q^\perp$ close to $\mathbf{c}_0$. Then $\mathbf{s} = \mathbf{c}_0 - \mathbf{v}$ is a valid signature:
  $$\mathbf{s}\mathbf{A}^T = \mathbf{c}_0\mathbf{A}^T - \mathbf{v}\mathbf{A}^T = \mathbf{c} - \mathbf{0} = \mathbf{c}. \tag{28}$$
  If $\mathbf{c}_0$ and $\mathbf{v}$ are close enough then $\mathbf{s}$ will be short, fulfilling the second requirement of a valid signature.

The GGH and NTRUsign schemes, however, proved insecure as the method for computing vector $\mathbf{v} \in \Lambda^\perp$ leaked information about secret basis $\mathbf{B}$ and have since been proven to be insecure by cryptanalysis[89–91]. The GPV framework differs in that instead of the deterministic algorithm - Babai's roundoff algorithm - it uses Klein's algorithm[75], a randomised variant of the nearest plane algorithm also developed by Babai[92]. Whereas both deterministic algorithms would leak information

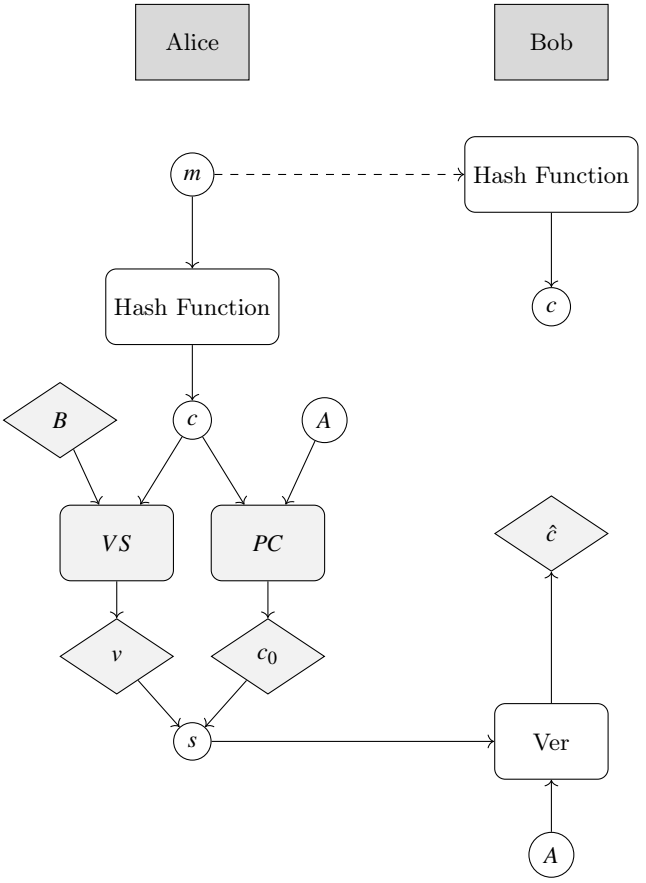

Figure 4: A schematic of the GPV signature framework. Alice hashes the message to a lattice vector $\mathbf{c} = H(m)$. Following this, she creates the key pair $(pk, sk) = (\mathbf{A} \in \mathbb{Z}_q^{n\times m}, \mathbf{B} \in \mathbb{Z}_q^{m\times m})$ and sends the public key to Bob. Using elementary techniques, she computes the preimage vector (PC) $\mathbf{c_0 A} = \mathbf{c}$, then using the secret key she samples the vector (VS) $\mathbf{v} \in \Lambda^{\perp}$ such that it is very close to $\mathbf{c_0}$. The signature is $\mathbf{s} = \mathbf{c_0} - \mathbf{v}$. Bob has to check the signature is small enough and, using the public key, that $\mathbf{s}\mathbf{A}^T [= \mathbf{c_0}\mathbf{A}^T - \mathbf{v}\mathbf{A}^T] = H(m)$. Here $\mathbf{v}\mathbf{A}^T = \mathbf{0}$ since $\mathbf{A}$ generates the lattice and $\mathbf{v}$ belongs to the dual lattice.

about the geometry of the lattice, Klein's avoids this by sampling from a spherical Gaussian over the shifted lattice $\mathbf{c_0} + \Lambda$.

Klein's algorithm was the first of a family of lattice algorithms known as trapdoor samplers. The GPV framework has become a generic framework into which a choice of trapdoor sampler and lattice structure can be inserted. It has also been proven to be secure under the assumptions of SIS under the random oracle model[86,93].

A prudent example of an instantiation of the GPV framework is the NIST submission FALCON[94]. This uses a trapdoor sampler known as the Fast-Fourier-Sampler - developed by Prest and Ducas[95] - over NTRU lattices that take advantage of the ring structure.

4. *Bai-Galbraith Signatures*

The Bai-Galbraith signature scheme[96] is an adaptation of an earlier work by Lyubashevesky[97] in which he develops a lattice-based signature scheme using a paradigm which he calls 'Fiat-Shamir-with-aborts'. Informally, it follows the chain of reductions:

$$\begin{array}{ccccc} \text{Hard Problem} & \leftarrow & \text{CRHF} & \leftarrow & \text{One-time signature} \\ & & & & \uparrow \\ \text{Signature} & & \longrightarrow & & \text{ID Scheme} \end{array}$$

Here CRHF stands for collision resistant hash function. The main idea is that, from a lattice based CRHF, one can create a one-time signature following Lyubashevsky and Micciancio[98], however this leaks information about the secret key. This would not be a problem for a one-time signature as the information becomes defunct after the usage. Constructing the ID scheme requires the repeated use of this, which is where the aborting technique comes in. In response to the challenge from the verifier, the prover can decide that sending the usual response would leak information and instead abort the protocol and restart it. The end result is a secure, if somewhat inefficient, ID scheme. Having to restart the protocol every time there is information leaked is done away with when adapting this to a signature scheme using Fiat-Shamir, however. The lack of interaction means that the prover can simply rerun the protocol until they find a signature which does not leak information.

As an LWE instantiation, Lyubashevesky's scheme has public key $(\mathbf{A}, \mathbf{b} = \mathbf{Az} + \mathbf{e} \mod q)$, where the components are picked as described above. The verifier picks small-normed vectors $\mathbf{y}_1$, $\mathbf{y_2}$ and computes $\mathbf{v} = \mathbf{Ay}_1 + \mathbf{y}_2$. With the message m they compute the hash $\mathbf{c} := H(\mathbf{v}, m)$ and the following two vectors: $\mathbf{s}_1 = \mathbf{y}_1 + \mathbf{zc}$ and $\mathbf{s}_1 = \mathbf{y}_2 + \mathbf{ec}$. The signature is then $s = (\mathbf{s}_1, \mathbf{s}_2, \mathbf{c})$. Here, to ensure that neither $\mathbf{s}_1$ or $\mathbf{s}_2$ leak information about the protocol, rejection sampling is employed. Developed across[99–101] this technique allows the vectors to be picked from a distribution independent of the secret. Verification requires checking that $\|\mathbf{s}_1\|$ and $\|\mathbf{s}_2\|$ are small enough, and that $H(\mathbf{As}_1 + \mathbf{s}_2 - \mathbf{bc} \mod q, m) = \mathbf{c}$. This can be thought of as a proof of knowledge of $(\mathbf{z}, \mathbf{e})$.

Bai and Galbraith adapted this such that it instead becomes a proof of knowledge of only $\mathbf{z}$ using a variation of the verification equation and compression techniques. Once the public key has been created, only one vector is required, $\mathbf{y}$, from which $\mathbf{v} = \mathbf{Ay} \mod q$. The least significant bits of $\mathbf{v}$ are then thrown away and the remainder is hashed with the message m to get a hash value c. From this value, the vector $\mathbf{c}$ is created and the signature is $(\mathbf{s} = \mathbf{y} + \mathbf{zc}, c)$, with rejection sampling again being used to check that the distribution of $\mathbf{z}$ is independent of the secret. Computing $\mathbf{w} = \mathbf{As} - \mathbf{bc} \equiv \mathbf{Ay} - \mathbf{ec} \mod q$ allows for verification by checking that the hash value of the

most significant bits of $\mathbf{w}$ with m is equal to c. See figure 5 for a schematic for the Bai-Galbraith signature scheme.

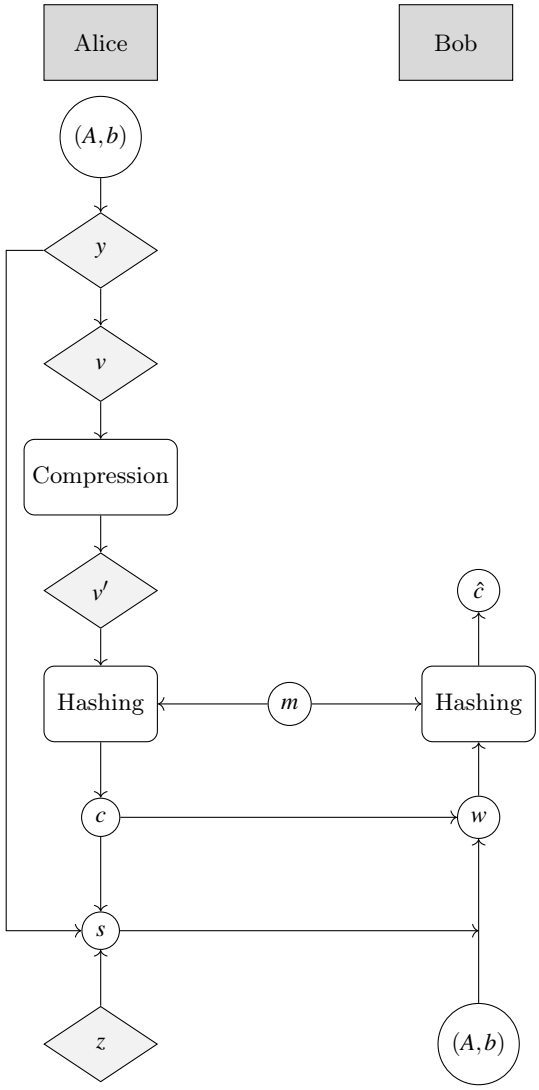

Figure 5: A schematic for Bai-Galbraith Signatures. Alice first generates the key pair $(pk, sk) = ((\mathbf{A}, \mathbf{b} = \mathbf{Az} + \mathbf{e} \mod q), \mathbf{z})$, sending the public key to Bob. After picking a short vector $\mathbf{y}$, she finds the finds the lattice vector $\mathbf{v} = \mathbf{Ay}$ and removes the most significant bits to compute $\mathbf{v}' = comp(\mathbf{v})$. She then hashes this with the message to create the vector $\mathbf{c}$ from hash value $c = H(\mathbf{v}', m)$. The signature is $\mathbf{s} = \mathbf{y} + \mathbf{zc}$. Bob computes the vector $\mathbf{w} = \mathbf{As} - \mathbf{bc}$ and checks that the hash value of the most significant bits of $\mathbf{w}$ and the message m is consistent with c.

The NIST submission CRYSTALS-Dilithium is based on Bai-Galbraith signatures over module-LWE.

5. Attacks

Similar to the multivariate case, there is a breadth of literature on specific attacks, both quantum and classical, and suffice to say we will not be going into them in too much depth. Here we give a brief summary of some of the avenues that have been attempted and references for readers to investigate. Similar to attacks on Multivariate Cryptographic schemes, these can largely be broken down into three categories: attacks on the underlying hard problems, attacks on specific schemes and side-channel attacks (see section III E for details of side-channel attacks).

a. General structure attacks The security of lattice cryptography can be largely reduced down to the shortest vector problem, so the general motivation for these attacks is to solve said problem. Schemes of this kind tend to come in two forms: algorithmic or sampling. Largely, the question to be answered is the following: given a random basis for a lattice, can one find the shortest vector? Or at least a small enough vector, satisfying the approximate SVP? On the algorithmic side, there are lattice reduction algorithms such as Schnorr[102] or Lyu et. al.[103] which attempt to find almost-orthogonal bases from a highly non orthogonal bases. In a similar vein, quantum speed ups of vector enumeration have been proposed by Aono et. al.[104] There are also search approaches such as the sieving algorithms[105] and their quantum counterparts[106]. Newer approaches, devised using adiabatic quantum computing, posing SVP as an energy minimisation problem as in Joseph et. al.[107,108] have also been developed. Similar to the use of sampling to generate small signatures, if a truly efficient discrete Gaussian sampler was developed it could pose a major problem[109,110]. Simply setting the centre of the distribution as the zero vector would allow the shortest vector to be picked with a high probability. There have also been suggestions that certain quantum algorithms could be used to pick vectors more efficiently from these distributions.

As mentioned previously, the SVP can be shown to be equivalent to solving the Hidden Subgroup Problem for symmetric groups, which has also been the focus of much quantum cryptanalytic research[111].

b. Specific attacks For details on how individual schemes are taking known attacks into account, we refer the reader to the design documents: Dilithium[112], FALCON[113].

D. Symmetric Primitive Based Submissions

Here we give a brief overview of the remaing two NIST submissions, Picnic and SPHINCS$^+$, both of which are based on symmetric primitives.

1. Picnic

Unlike the previously mentioned schemes, Picnic only requires the hardness provided by symmetric primitives such as hash functions and block ciphers[114]. It is a general scheme for the adaptation of a three-move proof-of-knowledge scheme (known as Σ-protocols) to signature using a transformation from Unruh[115]. It is claimed by Unruh[116] that the Fiat-Shamir paradigm for transforming proof-of-knowledge schemes into signatures is impractical to prove secure in the quantum random oracle model and so Unruh provides an alternative. The Picnic protocol provides two signature schemes: one via Unruh and another using Fiat-Shamir.

Picnic is built upon a Σ-protocol called ZKB++[114], which itself is built on an earlier scheme called ZKBOO[117]. For the sake of brevity, the details of Picnic and the underlying schemes are omitted and instead we will explain the underpinning framework by first explicitly defining Σ-protocols followed by Unruh's transformation.

Definition 29 (Σ-protocol). A three-move proof-of-knowledge protocol between a prover (Alice) and a verifier (Bob) is known as a Σ-protocol. Alice wants to prove she knows x such that $f(x) = y$, where y is commonly known, for some relation f.

1. Alice commits herself to randomness by picking r, which she sends to Bob.
2. Bob replies with a random challenge c.
3. Alice responds to the challenge with a newly computed t.
4. Bob accepts that Alice has proven the knowledge if $\phi(y,r,c,t) = 1$ for some efficiently computable and agreed upon ϕ.

Unruh's transform takes a given Σ-protocol with a challenge space C, an integer N, message m and a random permutation G and requires the following:

1. Alice runs the first phase of the Σ-protocol N times to produce $r_1, \ldots, r_N$.
2. For each $i \in \{1, \ldots, N\}$, and for each $j \in C$, she computes the responses t_{ij} for r_i and challenge j. She then computes $g_{ij} = G(t_{ij})$.
3. Using a known hash function, she computes $H\left(x, r_1, \ldots, r_N, g_{11}, \ldots, g_{N|C|}\right)$ to obtain indices $J_1, \ldots, J_N$.
4. The signature she outputs is then $s = \left(r_1, \ldots, r_N, T_{1J_1}, \ldots, T_{NJ_N}, g_{11}, \ldots, g_{N|C|}\right)$.

Bob then verifies the hash, verifies that the given t_{iJ_i} values match the corresponding g_{iJ_i} values, and that the t_{iJ_i} values are valid responses with respect to the r_i values.

Whilst Picnic is based on the Σ-protocol ZKB++, there is some choice in the use of the symmetric primitives used in the construction. The choice that has been implemented in the block cipher family LowMC[118], which is based on a substitution permutation network. Performance wise, Picnic is relatively slow and employs large signature sizes, but makes up for this with provable quantum security.

2. SPHINCS$^+$

SPHINCS$^+$[119] is based on an earlier protocol called SPHINCS[120]. This protocol is a hash-based, stateless signature scheme which had the goal of having the practical elements of other hash-based schemes and adding extra security by removing the stateful nature. A stateful algorithm depends in some way on a quantity called the state, which is initialised in some way. This is often a counter, though not necessarily, and stateful schemes can lead to many insecurities as they need to keep track of all produced signatures.

SPHINCS expands the idea of using a Merkle tree[30] to extend a one-time signature into a many-time signature scheme by creating a hypertree. In this tree of trees, leaves of the initial Merkle tree become the root one-time signatures for further trees, which themselves cascade into further trees. The size of the overall tree becomes a compromise between security and efficiency in the original scheme: which leaves are picked to become the next tree are chosen randomly, and so a smaller tree has a chance to repeat the leaf choice. In order to abate this, a few-times signature is used at the bottom of the tree. This randomness is what makes SPHINCS stateless. Whereas Merkle's original design iterates over the signing keys, SPHINCS builds on the theoretical work of Goldreich[121], in which the keys are picked randomly. The size of the hypertree allows the assumption that the new key has not been used before.

Improving on the previous design, SPHINCS$^+$ uses a more secure few-time signature at the bottom of the tree, known as Forest of Random Subsets (FORS), an improvement on a previous signature called HORST[120]. A better selection algorithm for choosing the leaves of the tree is also included. It also introduces the idea of tweakable hash functions.

Definition 30 (Tweakable hash functions.). Let $\alpha, n \in \mathbb{N}$, $\mathscr{P}$ be the public parameters space and $\mathscr{T}$ be the tweak space. A tweakable hash function is an efficient function

$$TH : \mathscr{P} \times \mathscr{T} \times \{0,1\}^{\alpha} \to \{0,1\}^{n}, \quad MD \leftarrow TH(P,T,m), \tag{31}$$

mapping an α-bit message m to an n-bit hash value message digest (MD) using a function key called public pa-

rameter $P \in \mathscr{P}$ and a tweak $T \in \mathscr{T}$.

This allows the hash functions to be generalised to the whole hypertree as they can adapt to changes in the chosen leaves of the sub trees. In brief, the SPHINCS$^+$ signature scheme can be summarised as follows:

1. Alice generates $p, q \in \{0,1\}^n$: p is a seed for the root of the top tree in the hypertree and q is a public seed. The pair (p,q) form the public key. The secret key is the pair $t, u \in \{0,1\}^n$, respectively seeds for the few-time signature FORS and the chosen one-time signature for the protocol, WOTS$^+$[119].
2. To sign the message, Alice generates the hypertree and the signature is the following collection: a FORS signature on a message digest, a WOTS$^+$ signature on the FORS public key, a series of authentication paths and WOTS$^+$ signatures to authenticate the the WOTS$^+$ public key.
3. To verify this, Bob iteratively reconstructs the public keys and root nodes until the top of the SPHINCS$^+$ hypertree is reached.

The SPHINCS$^+$ protocol was not designed with the same kind of performance in mind as either the lattice or multivariate schemes. Generally, stateless signatures have much larger key and signature sizes, as well as slower performances. They mainly target applications which have low latency requirements but very strong security requirements, such as offline code signing. In Bernstein et. al.[119] the reader will find an analysis of the security of this scheme in both a classical and quantum setting that shows it to be very strong in both regards.

3. Attacks

Here we include references for cryptanalysis efforts of the above schemes. For Picnic the recent attacks include a multi-attack on the scheme and it's underlying zero-knowledge protocols[122], and an attack on the block cipher used in implementation, LowMC[123]. Both of these - as well as some side-channel attack analysis - are addressed in the design document[124].

Currently, the main attack against the SPHINCS framework is that of Castelnovi, Martinelli and Prest[125,126]. This is a type of side-channel attack known as a differential fault attack. In the design document[127], general protection against this kind of attack - as well as other known general attacks - is addressed.

E. Side-Channel Attacks

It would be remiss of this paper to give an overview of the state of contemporary cryptographic signatures - especially with regards to new standards - without a note on side-channel attacks. Side-channel attacks are cryptanalytic attacks that focus on finding flaws in the implementation of protocols rather than the design. Examples of this include timing attacks - where an adversary can gleam information about a secret from a protocol by taking advantage of a subroutine running in non-constant time- and energy attacks - a similar process but instead requires examining the energy use of the protocols. This has led to some of the bigger breaks of security systems that have been employed[128] (p. 116). Unfortunately, in the case of Post-Quantum Cryptography, it is often a form of attack which has not been considered in as much depth as is potentially necessary and many of the submissions are missing a large scale analysis of how they could be affected. Similar to the attacks previously mentioned, however, it is beyond the scope of this paper to go into a great amount of detail so instead we provide a list of references for invested readers.

For a good overview of side-channel attacks in general, we refer the readers to Fan et. al.[129] or Lo'ai et. al.[130] Beyond this, we direct the readers towards specific analysis of side channel attacks for certain NIST submissions:

There has been some work on general fault attacks on Multivariate public key cryptosystems[131]. Side-channel attacks on Rainbow can be seen in the general attacks on Unbalanced Oil and Vinegar schemes[132–134].

Dilithium was attacked using a side-channel assisted existential forgery attack[135]. This was responded to with countermeasures suggested in the implementation that mask the protocol[136]. FALCON has recently been attacked using a protocol known as BEARZ[137]. The attacking party also suggested countermeasures to prevent this fault attack and timing attacks on FALCON.

For each submission, we also direct the readers to the respective design documents[52,112,113,124,127,138]. Unfortunately the level of detail on each is not to an equal standard, with some severely lacking side-channel attack analysis.

F. Performance

When considering the performance of these different protocols, there are various angles to analyse them. At a top level view one could compare a range of properties such as the key size, length of signatures, verification times and signature creation times. NIST make the point that these algorithms will be employed in a multitude of applications, each with different requirements. For example, if the applications can cache public keys, or refrain from transmitting them frequently then the size of the public key is not as important. Similarly, in terms of the computational efficiency, a server with high traffic spending a significant portion of its resources verifying

client signatures will be more sensitive to slower key operations. The call for proposals[49] even suggests that it may be necessary to standardise more than one algorithm to meet the differing needs.

Whilst the computational efficiency relies on the specific architecture used, the key and signature sizes can be compared theoretically. A comparison of the NIST schemes for these architectures can be found in table III. Many of the submissions include data on several variants of their respective schemes but we have only included a cut down list here. The variants - as well as the original data - can be found in the submissions themselves[52–57].

Submission	PK Size	Signature Size	Security level
GeMSS 128	352188	32	1
GeMSS 192	1237964	51	3
GeMSS 256	3040700	72	5
Rainbow Ia	148500	32	1
Rainbow IIIc	710600	156	3/4
Rainbow Vc	1683300	204	5
Dilithium 1024×768	1184	2044	1
Dilithium 1280×1024	1472	2701	2
Dilithium 1760×1280	1760	3366	3
FALCON-512	897	657	1
FALCON-768	1441	993	2/3
FALCON-1024	1793	1273	4/5
Picnic-L1-UR	32	53961	1
Picnic-L3-UR	48	121845	3
Picnic-L5-UR	64	209506	5
SPHINCS$^+$-128s	32	8080	1
SPHINCS$^+$-192s	48	17064	3
SPHINCS$^+$-256s	64	29792	5

Table III: Comparison of the key and signature sizes of the NIST round 2 signature submissions. All sizes given to the nearest byte.

The timings, however, do have a certain dependence on implementation and as such NIST have set out their requirements with respect to the NIST PQC reference platform[49]: an Intel x64 running Windows or Linux and supporting the GCC compiler. A comparison of the performance of the timings of the schemes can be found in table IV. Unless noted otherwise, the submissions used the reference architecture.

G. Concluding Remarks on Post-Quantum Digital-Signatures

Whilst progress in the field of quantum computing does pose a threat to our current digital security models and implementations of digital signature schemes, throughout this section we have given a brief overview of the work being done to combat this direct threat. Working towards NIST's criteria for both security and efficiency ensures that sought-after solutions are

Submission	Key Gen	Signing	Verification
GeMSS 128	38500	750000	82
GeMSS 192	175000	2320000	239
GeMSS 256	532000	3640000	566
Rainbow Ia	35000	402	155
Rainbow IIIc	340000	1700	1640
Rainbow Vc	757000	3640	239
Dilithium 1024×768	243	1058	273
Dilithium 1280×1024	371	1562	376
Dilithium 1536×1280	471	1420	511
FALCON 512	6.98*	$6081.9^{\dagger}$	$37175.3^{\ddagger}$
FALCON 768	12.69*	$3547.9^{\dagger}$	$20637.7^{\ddagger}$
FALCON 1024	19.64*	$3072.5^{\dagger}$	$17697.4^{\ddagger}$
Picnic-L1-UR	160	172560	116494
Picnic-L3-UR	392	549036	368492
Picnic-L5-UR	753	1234713	828446
SPHINCS$^+$ 128s simple	326805	4868849	5304
SPHINCS$^+$ 192s simple	486773	10259965	7971
SPHINCS$^+$ 256s simple	636421	7570079	10866

Table IV: Comparison of the signature creation and verification times of the NIST round 2 signature submissions. Unless stated otherwise these are all to the nearest thousand processor cycles. *milliseconds, † signatures/second, ‡ verifications/second.

implementable with current technology, well ahead of implementations of Shor's algorithm being practical. Where theoretical protocols have struggled to reach a compromise between security and efficiency, research has already yielded results to adapt, as shown by modifications of the UOV-schemes and SPHINCS.

Indeed, this is the case in reality also, with Google implementing a lattice-based protocol in their Chrome web-browser (although this has since been removed in an effort to 'not influence the standardization procedure')[139]. However, we remain wary of further threats presented. For example, that a solution to the HSP problem could cause issues for lattice cryptography showcases the need for research to continue remaining ahead of the curve.

The National Institute for Standards and Technology's goal of implementing a new standard is predicted to still be at least a year off completion so it is far from finalised. It is certainly likely that the recommendation will be several new standards depending on the application. Whilst there is a notion of simplicity leading to security, some of the above schemes eschew this in favour of ruthless efficiency (albeit with the necessity for careful implementation), with some even claiming to be faster than current protocols. Ultimately the advances in cryptography in response to quantum computing appear to be leading to altogether more complicated systems, but for the sake of security this is certainly the right move.

IV. QUANTUM DIGITAL SIGNATURES

A. Key Concepts General to QDS

The security of classical digital signatures lies in creating problems that are infeasible to solve. The security of techniques based on quantum physics instead relies upon proven scientific principles[140]. The uncertainty inherent in quantum physics has in recent years found a great many uses in the field of security, ranging from random number generators to optical identity tags[141]. This well known phenomena is what protects a system against attack, as in many cases, it is physically impossible for the attacker to breach the system without detection.

As with many classical digital signatures, quantum digital signatures rely upon a one way function for their encryption. In this case however, rather than using a mathematical one way function, classical data is encoded as quantum information[142]. In order to implement this each quantum digital signature protocol follows three steps similar to those in classical cryptography[143]:

1. GEN: Alice uses her private key to generate a signature s consisting of quantum information.
2. SIGN: Alice sends her message m to the recipients (denoted Bob and Charlie) with the corresponding signature, denoted as (m, pk).
3. VER: Bob and Charlie verify the message is authentic and repudiation has not occurred. In the general case this involves a comparison of (m,s) to the classical description of s.

In order to delve further into what each step entails we will discuss a generic QDS model, although schemes will vary in their specific implementation of these steps most follow this general framework[143][142].

The generation step begins with a purely classical operation, the random generation of a private key for each possible single bit message (1 or 0). This key, denoted as $pk^i = (pk^i_1, pk^i_2, ..., pk^i_L)$, is purely classical information, where $i = 0, 1$ to denote the message. Its length L is determined by the level of security required and the QDS scheme used. It is using this string that Alice will identify herself at a later stage so it is imperative it is never shared.

The next stage of the generation step is done by first defining a set of non-orthogonal quantum states[143]. An example of quantum states that can be used is the BB84 states[144]. Alice then generates four separate strings of quantum information (known as quantum digital signatures) by encoding her pk strings using the defined quantum states. These four signatures consist of a copy of the encoded private key for both possible messages for both Bob and Charlie. These are denoted as qs^i_B, qs^i_C with the subscripts denoting who the signature pairs will be sent to. Alice then sends qs^i_B, qs^i_C to the correct recipient via a secure quantum channel. Bob and Charlie measure their quantum signature pairs to generate a classical signature from them, s^i_B, s^i_C .

Before proceeding to the next step in most cases Bob and Charlie randomly select approximately half of the elements in their measured signature (though this can occur before measurement) and forward to the other. They do not have to exchange the same elements. As such to all extents and purposes from Alice's perspective both s^i_B and s^i_C are exactly the same. This prevents her from committing repudiation. The exact method of this "symmetrisation" is dependant on the QDS scheme.

To sign a message in most protocols Alice sends her message of a bit of 1 or 0 with the corresponding private key to Bob[145], denoted (m^i, pk^i). As the protocols focus only on the signing of a message it is assumed that the message is sent along a secure channel whether this be quantum or classical in nature[143]. To send a multi-bit message the process of generation and signing is iterated for each bit[146].

Finally there is the verification stage. This varies greatly between each protocol (see relevant protocol section for specific details). The general case is that Bob compares his measured signature s^i_B to the private key pk^i received from Alice. If the number of mismatches between the two is below the required threshold (see section IV A 1) Bob deems the message as authentic. If Bob wishes to forward the message to Charlie, he sends the (m^i, pk^i) he received from Alice. Charlie then performs the same process with a different required threshold.

The security of quantum digital signatures is shown most prominently in the validation step. Each of the signing protocols relies on the same principles to provide security[143]. It is key to this that the states chosen are non-orthogonal. Therefore, any measurement performed on one state will not commute with a measurement on the other[140]. Thus, any measurement performed will probabilistically disturb the other state, effectively destroying the information it held[147]. As such without knowing the initial private key no one can discern the original classical input. Getting the correct result is entirely dependent on chance, even then one would have no way to tell if it is the correct result[142].

This is reinforced by the Holevo bound placed on the information that can be obtained, only a single bits worth of information may be extracted from a qubit[148]. This prevents further information to help the attacker's deductions from being obtained[142]. If anyone attempted to forge a signature they would have to guess the private key correctly based on the information they can gather.

For short private keys this is indeed improbable but still possible. As a signature gets longer however the chance of successfully guessing falls off exponentially to a negligible value for a long enough private key[143][142]. As such a simple comparison will reveal their ruse[149]. Therefore unlike classical digital signatures QDS is not dependant on the difficulty of mathematical techniques and as such demonstrates information theoretic security[140].

1. *Quantifying Authentication*

There are three possible levels to the degree of verification that Bob and Charlie can deem the message/signature combination has fulfilled[142]:

- 1-ACC: Message is valid, can be transferred.
- 0-ACC: Message is valid, might not be transferable.
- REJ: Message is invalid.

1-ACC and 0-ACC provide similar levels of security in the standards they uphold, both require that the message be valid. As such if Bob performs the validation test on the signature and finds it to be true then the first condition of these security levels are fulfilled. The difference arises in the transferability of the signature. If Bob has any reason to believe that Charlie would not come to the same conclusion as him then the message and signature would be deemed 0-ACC (i.e. repudiation has occurred). Only a message with validation level of 1-ACC is deemed fully secure. Finally if the signature that Bob receives is invalid then the message is given the validation level REJ and is rejected.

These criteria are mathematically defined by a series of thresholds proposed by Gottesman and Chuang[142]. We first define the probability of failure of an attacker p_f, this is the probability their measurement will fail to get the correct classical description given a minimum error measurement. There is also the probability that an honest measurement will fail due to environmental factors such as noise, denoted p_e. Both of these thresholds are calculated based on outside factors such as the number of copies of the signature circulated, the method with which measurements are taken, the apparatus used to distribute/measure the signatures, etc. Using these as the boundaries for acceptable error levels, with p_f as the upper and p_e as the lower, the allowed error thresholds can be set[142]. For a signature to be authenticated it must have a fractional error lower than s_v, wherein $p_f > s_v$. Based on this the upper bounds of the probability that the forger can effectively mimic a valid signature is given by: $e^{-c(p_f - s_v)^2 L}$[143], where c is a constant. This shows that the security against forging is dependant not only on the environmental factors (as p_e and p_f define the range in which s_v can fall) but on the length of the signature itself[143].

One threshold is not enough to protect against repudiation as stated by Gottesman and Chuang[142]. If Alice was dishonest and sent different signatures to Bob and Charlie (with the aim to have Bob accept and Charlie reject it), she could successfully repudiate with only one threshold. With only one threshold (s_v) for verifying the signature Alice could tailor each signature to have $s_v L$ errors. The number of errors is defined by a normal distribution with a mean of $s_v L$. As such in the probability of Bob accepting the message (errors below $s_v L$) and Charlie rejecting it (errors above $s_v L$) is 0.25[143]. By introducing a second threshold s_a where $s_v > s_a$ that Charlie must pass instead of s_v reduces the probability of successful repudiation to negligible with a long enough signature. This is due to the fact that Alice would have to generate a signature that would give a result both below s_v and above s_a. In a similar manner to forgery the upper bound on the probability for successful repudiation is now defined by $e^{-c'(s_a - s_v)^2 L}$, where c' is a constant.

This defines the necessary criteria for setting out mathematically how to achieve the verification conditions. To summarise the relative size of each of the thresholds: $0 < p_e < s_a < s_v < p_f$. In order to achieve a 1-ACC level of verification a signature's error fraction must be below s_v for Bob and s_a for Charlie. Falling below only s_v would result in a 0-ACC rating and falling below neither results in a REJ.

B. QDS with Quantum Memory

The first QDS protocol was proposed by Gottesman and Chuang in 2001[142], hence referred to as GC-QDS. As the precursor to all other QDS protocols as expected GC-QDS was only a theoretical proposal, however it is important to analyse. It sets the standard for the "ideal" quantum digital signature protocol. One in which the signature remains as quantum information throughout the entire process. This ensures information theoretic security throughout.

As detailed in section IV A, to begin with Alice generates her random private keys, pk^i, for each possible message value. The initial proposal for this scheme suggested each private key element, pk^i_n, be a two bit string and qs^i_n the corresponding BB84 state[142]. Using this, Alice generates copies of each quantum digital signature for each possible message value by encoding pk^i as quantum information. As many copies of the public key as there are recipients are generated and one distributed to each. The recipients, in this case Bob and Charlie, then do not measure the received signature and instead store it in stable quantum memory, a key difference to other protocols. This is the "generation"

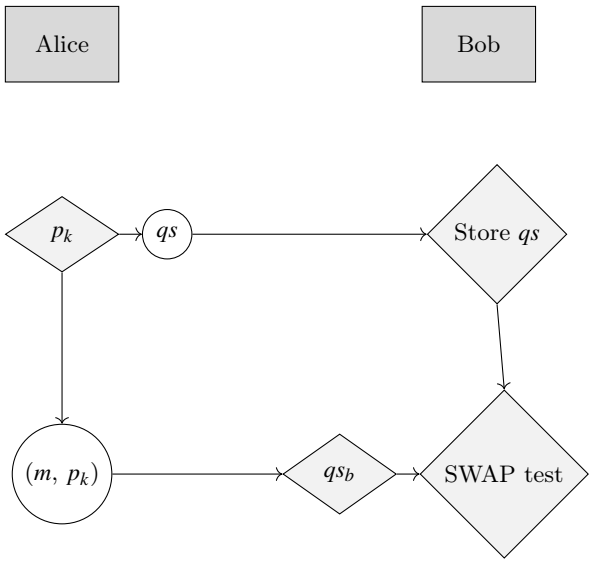

Figure 6: Flow diagram breaking down the process of Gottesman and Chuang's first proposal for a quantum digital signature protocol. Alice begins by randomly generating her private key (p_k). From this she generates her quantum signature (qs) which is sent on to Bob. Bob stores this in quantum memory without measuring it. To sign a message (m), Alice sends it alongisde its corresponding p_k to Bob. Bob uses p_k to genreate his own copy of the quantum signature qs_b. He verifies its authenticity in a SWAP test with qs. Diamonds represent information that must be kept private (at least until sending), circles represent information that is sent to another individual. In the case of this QDS scheme the public key is quantum information, all other information is classical.

phase for this protocol[142].

For the signing phase, Alice only sends out classical information. She sends to Bob (m^i, pk^i). As with most QDS protocols this is assumed to be performed over a secure classical channel. An assumption that can be inexpensively implemented and so is not outlandish to assume.

Finally there is the verification phase. Using pk and the known function for encoding classical information as quantum information Bob generates his own set of quantum states. He then compares the states he has generated to those he has stored from Alice using a SWAP test[142]. If the number of mismatches between the two falls below the acceptance threshold s_a, Bob accepts the message as authentic. For further verficiation he can forward (m^i, pk^i) to Charlie who will repeat the process with his threshold s_v.

The unforgeability in this scheme stems from the strict policy of not measuring the quantum digital signature until authentication is required[143], as demonstrated in figure 6. For an attacker to forge a signature they would as in, classical digital signatures, have to bypass the one-way nature of of the classical to quantum encoding. Aside from obtaining the private key from Alice the only way to achieve this is to intercept the quantum digital signature and correctly guess the measurement basis for each qubit. Thus, as previously discussed, for a long enough signature there is a negligible probability that Bob and Charlie will not notice that the signature has been tampered with[149]. Owing to the collapse of a quantum wavefunction upon measurement, the very act of attempting to intercept and forge a message will reveal an attack has occurred as the signature remains only as quantum information. The only attack that can be achieved by interception is to cause the qs distribution phase to abort[142].

The protection against repudiation lies with the comparative SWAP tests and relevant thresholds that Bob and Charlie apply to their stored public keys[142]. As SWAP tests do not measure a quantum state and instead compare two to determine similarity it makes them the perfect operation to enforce non-repudiation. Coupling the non-destructive nature of the SWAP test with the thresholds detailed in section IV A renders the probability of repudiation to negligible with a long enough signature[142].

This scheme is not without faults however, its most prominent fault is its reliance on the immature technology of quantum memory. If the technology were perfected it would allow for the indefinite storage of qs. As of time of writing however, quantum memory can not store quantum information for long time periods[150][151]. In theory this protocol can have long term quantum digital signatures but in practice this is simply not possible. Although there have been recent advancements in the storage times of quantum memories this protocol is currently infeasible[145][152].

The SWAP tests themselves are an issue within the same vein as quantum memory, the technology to perform them is not available. Each recipient would require a quantum computer in order to perform such a test. As with the quantum memory requirement this ensures Gottesman and Chuang's theoretical proposal remains theory.

C. Multiport set-up

The attractive concept of quantum digital signatures coupled with the initial proposal's reliance on quantum memory and computing lead to a great deal of interest in QDS from both a theoretical and experimental standpoint[140]. The reliance on currently immature

quantum technologies has led to new proposals that find methods of getting around this constraint. One of the earliest proposals was that of the multiport[153]. This apparatus consists of a square array of four separate 50:50 beam splitters as shown in figure IV C[149]. The apparatus and its potential in cryptography was first proposed in 2006[147]but only as a theoretical method of public key distribution. Later, in 2012, it was adapted for use in quantum digital signatures and experimentally demonstrated.

For use in quantum digital signatures the multiport is effectively split in two, the top two beam splitters belong to Bob and the bottom two to Charlie. The multiport in of itself primarily affects the generation stage of the quantum digital signature process.

The generation stage begins the same as other QDS schemes. Alice generates a randomised classical private key which she encodes with her chosen quantum basis. She then sends these to Bob and Charlie. She sends the the copies of each quantum digital signature at the same time (i.e. qs_B^1 and qs_C^1 are sent out at the same time and then the other set of copies are sent).

The first set of beam splitters (moving left to right on figure 8) are used by Bob and Charlie to split their copies of the signature into two equal amplitude components. One half of these are kept by Bob/Charlie and the other half are sent to the other recipient. This ensures that Alice is unaware as to who has which bit in each of the copies that she initially sent[149151]. At the second set of beam splitters the half that was originally kept by the receiver is mixed with the half from the other recipient in a comparison test. The process of this is detailed in figure 7.

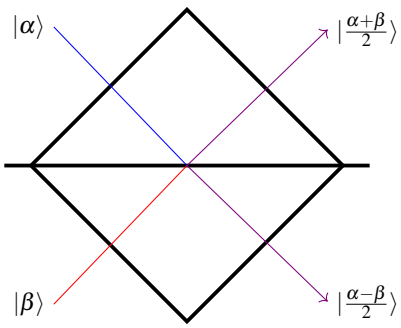

Figure 7: Schematic diagram of a single beamsplitter. From the left enters two separate photon packets, $|\alpha\rangle$ and $|\beta\rangle$. These are combined by the splitter to give $|\frac{\alpha+\beta}{2}\rangle$ and $|\frac{\alpha-\beta}{2}\rangle$.

In the initial implementation of this protocol, Bob and Charlie would store the received states in quantum memory ready for the verification stage. In 2014 Dunjko showed that it was indeed possible to remove the quantum memory from this protocol[149]. Progressing instead to verification via comparison of classical data, the latter will be the focus of this section.

Rather than storing the quantum digital signature, the incoming signature is measured and the outputs of this measurement stored. This classical string of results then forms an individual's measured signature. Most commonly the measurements performed are quantum state elimination measurements (covered in further detail in section IV D), however unambiguous state discrimination was initially proposed[151]. The key principle of either method is that it does not give a completely accurate description of Alice's initial private key. Thus, it does not enable a recipient to then forge a signature. The measurement and storage of this now classical signature completes the generation stage of the protocol.

To sign a message Alice simply sends her single bit message alongside the relevant private key to Bob[151]. Bob then compares the private key to the signature he measured, applying the s_a threshold detailed in section IV A 1 to determine the degree to which he trusts it. He then forwards the key and message to Charlie to validate that Alice indeed sent them both the same signature and similarly Charlie compares the two to see if the errors between them fall below s_v. From these results they determine the level of validity in the message and signature.

The security of this protocol lies with the square array of beam splitters[153]. It is the secondary beam splitters that Bob and Charlie use to check the validity of the signature. One output of the beam splitters will be a mixed state of $|\alpha\rangle$ and $|\beta\rangle$. If Alice was honest then $|\alpha\rangle = |\beta\rangle$ and as both are identical and coherent the initial input from Alice is obtained. If, however, Alice sent Bob and Charlie different signatures the multiport will symmetrize these and prevent repudiation[153]. The null ports then act as a safeguard against active forging[151]. In this scenario Bob is the dishonest party and attempts to forward a forged signature and message to Charlie. In this case Charlie would measure a non-zero (assuming no background count) reading on his null port, informing him to the presence of a forged signature.
One of the key issues with this scheme is the loss present in this system, measured at 7.5dB. The signature length L required for a security level of 0.01% was 5.0×10^{13} for a half bit[151]. The count rate with USE was found to be 2.0×10^5 counts per second. Given that this would yield a time required of 7.9 years to sign and send this is clearly an impractical signature length. This is particularly troublesome when one considers this was not taken at a separation distance, $5m$, great enough for use in a practical setting[151]. Methods of improving upon this technique were proposed such as increasing the clock rate of the VCSEL used to generate the pulses. Due to the loss

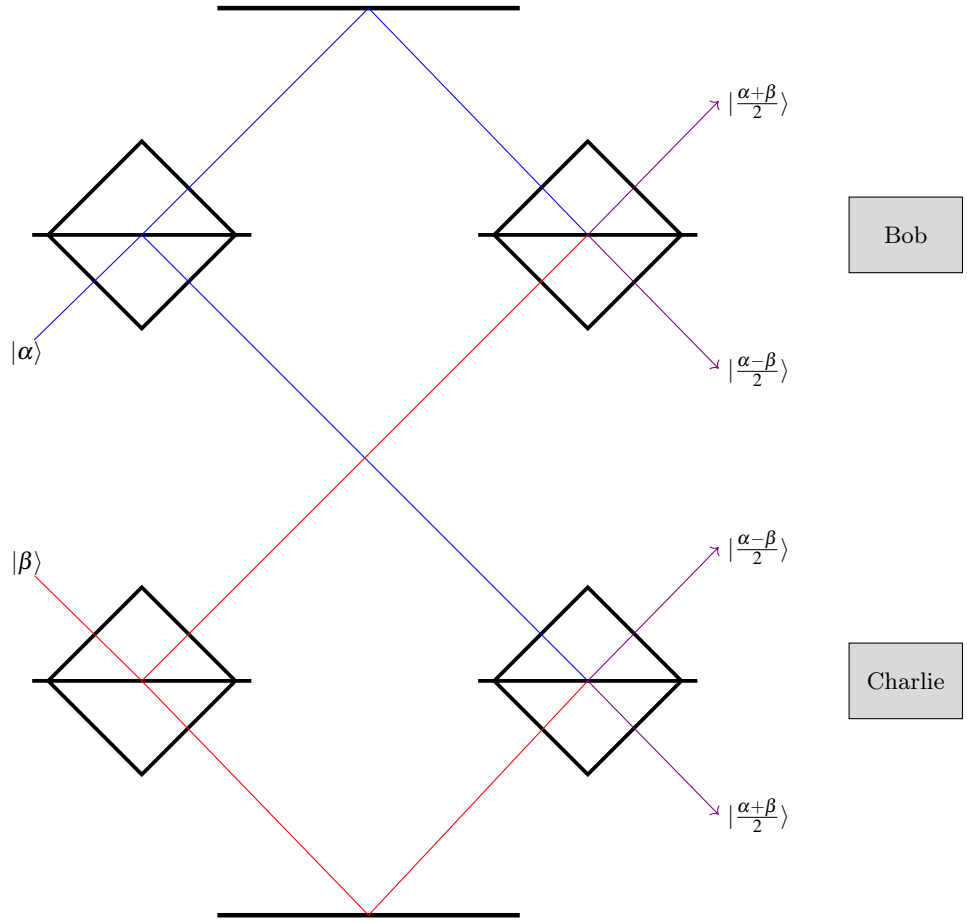

Figure 8: Schematic diagram of the multiport set-up used in the multiport QDS scheme. Each bisected diamond represents a beamsplitter. The thick black lines at the top and bottom of the diagram represent mirrors. One half of the array is in the posssesion of Bob and the other in the possession of Charlie.

rates and impractical distance requirements however, using multiports in quantum digital signatures serves only as a proof of concept for not requiring advanced quantum technologies.

D. Random Forwarding

The development of quantum digital signature schemes shows common themes, the reliance on quantum mechanics to achieve information theoretic security and moving away from the reliance on immature quantum technologies. Multiports, whilst flawed, demonstrated that quantum memory and quantum computing is not necessary for QDS. The next clear step, as stated in the paper implementing multiports, is to develop a system that does not use them for security[151]. The simplest way to achieve this is random forwarding.

As with all previous schemes, Alice begins by generating a string of classical bits that she keeps as her private key. She then encodes this in quantum states using non-orthognal bases[154]. Once again two copies of each quantum digital signature are created and the copies of the same signature are sent to Bob and Charlie at the same time (arrows 1 and 2 on figure 9).

Unique to this protocol is that upon receiving the quantum digital signature Bob and Charlie randomly choose elements of the signature qs^i to forward to the other, usually by a "coin toss" protocol[154145], arrow 3 on figure 9. They then record the location in the string of those that were passed on and which elements were retained. If either receive less than $L(\frac{1}{2}-r)$ or more than $L(\frac{1}{2}+r)$, where r is a threshold the two of them set, they abort[154]. Thus Bob and Charlie's final quantum digital signature will be a randomised mix, who's contents is defined only by the "coin toss" performed to dictate whether to keep an element. From the view point of Alice therefore once she has sent the messages the reduced density matrices for Bob's and Charlie's quantum digital signature elements are identical, regardless of whether or not she tried to commit repudiation[154].

Bob and Charlie then measure their quantum signature copies to get their measured classical signature s^i. This is known as the pre-measurement approach[154]. Alternatively Bob and Charlie can measure the quantum digital signature elements before forwarding in a post-measurement approach. In this case there is no need for a quantum communication channel between them, reducing the system to only having the quantum channels between them and Alice. In either scheme the technique of USE (Unambiguous State Elimination) is commonly used to measure qs[145]. Whichever approach is applied the random forwarding scheme is secure against forgery committed both by Bob/Charlie. In order to convince the other that a forged signature is valid they would need to correctly guess the measurement results of the half of the others signature that they did not receive. For a long enough signature the probability of this occurring is negligible.

To sign a message Alice simply sends her private key concatenated with the message to Bob (arrow 4 on figure 9). Bob verifies the authenticity of the signature by comparing how many signature elements he correctly eliminated. If the error in this falls below s_a then the message is deemed authentic and he passes it along with the key to Charlie. Thus, protecting against the possibility of an outside attacker forging a signature. Charlie performs the same comparison with his stored classical signature and his error threshold s_v. If it passes this then the message is deemed valid. Repudiation has not occurred due to the symmetrised signatures and Bob has not committed a forgery as Charlie's signature has successfully been compared with Alice's.

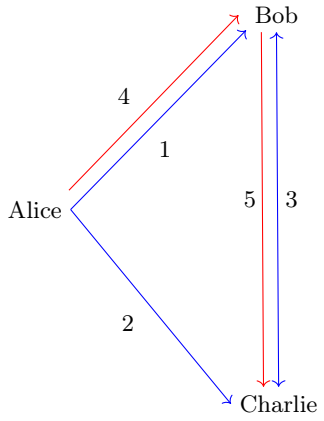

Figure 9: Representation of the communication channels between Alice, Bob and Charlie in QDS schemes based on random forwarding. Blue arrows represent quantum communication channels and red classical channels. Single headed arrows represent one way communication channels, double headed represent two way channels.

The benefits of random forwarding are apparent in the reduced dependence on quantum technologies. Stripping back QDS protocols so one is only using QKD to handle quantum information is far more practical than quantum memory or a multiport[145155]. The technology is more mature and has undergone a rigorous amount of field testing. Thus, allowing for easier integration of QDS into existing networks. In addition using QKD based quantum communication adds, in exactly the same manner for QKD, protection against message interception. Although this scheme would result in the sacrifice of some bits for Alice and Bob to compare it would remove the risk of outside interception[156]. Giving the random forwarding protocol a wide range of applications and versatility. As such many other schemes build on the primitive of random forwarding. Either by developing more advanced hardware and measurement techniques[157] or using it as the primitive for symmetrisation in schemes such as a QKD based schemes proposed by Collins et al.[155] detailed in section IV E.

E. QKD Key Generation Protocol

Quantum cryptography as a field did not begin with the development of QDS, but instead with a far more developed technique is that known as Quantum Key Distribution. This is a process by which, through the exchange of quantum information, Alice and Bob can generate a secret key for use in encrypted communication. By observing the error rates that occur in the measurement of the quantum information it is possible to detect if an eavesdropper is present[140]. Through this it can be confirmed if the exchanged key is indeed completely secret. If so by using a one time pad protocol Alice and Bob can achieve completely secure encryption. This in of itself could be used to generate a digital signature[154156]. As Alice and Bob would know that the signature could only be known by one of them it would act as an identifier.

This principle in of itself however, is not of particular use. Firstly it does not allow for more than one recipient, a major drawback for a signing scheme. Secondly it was shown that schemes based on partial QKD protocols could be expanded to multiple recipients and were more efficient in signing than secret key exchange via QKD[156].

The Quantum Key Distribution Key Generation Protocol (QKD KGP) differs from other quantum digital signature schemes in how the quantum information is distributed. It does, however, still follow the usual three step process of generation, signing and verification.

In the generation step, rather than Alice sending quantum information to Bob and Charlie, they send it to Alice[155]. This is done to simplify security analysis as it means Alice cannot send out entangled states.

As with Alice in other schemes Bob and Charlie first generate their own individual random classical private key for both possible message values (pk_B^0, pk_B^1 and pk_C^0, pk_C^1), recording the basis that each element was encoded in. They encode this as quantum information using the chosen method (both BB84 states and phase encoding have been used[155,156,158]). Thus, forming two sets of separate quantum digital signatures qs_B^0, qs_B^1 and qs_C^0, qs_C^1.

Bob (Charlie) then performs a partial QKD protocol with Alice[156]. He sends his quantum digital signatures to her, Alice chooses a random basis, based on the encoding method used, to measure each element. Resulting in Alice having two strings of classical elements from Bob (Charlie), s_B^0 and s_B^1 (s_C^0 and s_C^1). Bob (Charlie) then announces which basis each element was encoded in for each qs over a classical channel to Alice, who then "sifts" her signatures by discarding any that weren't measured in the same basis. Bob (Charlie) discards the corresponding elements of his private key. Two further sections of the measured signatures and corresponding private keys are then sacrificed[156,158]. At first Alice and Bob (Charlie) determine the hamming distance between corresponding sections of their signature and private key. If they are sufficiently correlated (they need not be exactly the same) the process proceeds and those sections are discarded. Secondly, Alice and Bob (Charlie) compare corresponding sections in order to observe if the error that an eavesdropper who induce in the measurements is present. If not these sections are discarded and the process continues. Each test is performed over a classical channel and if either of these steps are not successful the process repeats.

Alice will now have 4 sets of signatures, s_B^0, s_B^1, s_C^0 and s_C^1. As the error correction usually present in QKD protocols has not been performed these measured signatures will not exactly match their private key counterparts. Bob and Charlie then randomly select half of their private keys and forward them on to the other over a secret classical channel. To all extents and purposes from the perspective of Alice, each of these keys is identical, as she does not know what has been forwarded[155].

To send a message Alice then sends to Bob (m^i, s_B^i, s_C^i). To verify the authenticity of this, Bob compares s_B^i to his corresponding private key pk_B^i and s_C^i to the half that he received from Charlie. If the fraction of mismatches in both is less than s_a then he deems them authentic. For further verification he can forward (m^i, s_B^i, s_C^i) to Charlie who will repeat the process with s_v as his threshold.

The security of this scheme relies on the already proven security of QKD whilst adapting that pre-existing technology for use as a quantum digital signature[156]. The key difference to other QDS schemes, that of two different quantum signatures for each message, improves the efficiency of the scheme over both other QDS schemes and secret key sharing via QKD. Neither Bob nor Charlie can be dishonest and attempt to forge as they do not know the half of the other's private key that was not sent. This is opposed to other QDS schemes wherein a forger has access to the whole of the QDS. The different private keys removes the risk of colluding forgers whom in other schemes would have had a copy of the QDS each to try and determine the correct measurement values from it. The only option for a forger is to eavesdrop, which can be determined from the sacrificed bits. As well as this Alice cannot commit repudiation as she does not know who has which private key elements. Finally this scheme's lack of the need for error correction and privacy amplification as in a full QKD protocol means it is more tolerant to noise. As such it can be implemented in QKD based systems and used in a wider variety of scenarios.

F. Expanding to Signing Multiple Bits

The protocols detailed in this report have all focused on signing a single bit of data. The message is encoded into the quantum digital signature by having a signature for both possible message values. As of yet there is not a great calling to analyse multi-bit messages as the focus is on producing a practical single bit protocol. It is proposed that single bit signing protocols are expanded in a simple manner to sign a message of many bits[159]. For each bit in the message as a whole the signing process is iterated. Multiple different signature pairs would be sent to Bob and Charlie. When Alice sends her multi-bit message she would send each bit with a valid private key string. Nonetheless some papers have, however, raised concerns over this. Citing that insufficient research has been performed in this area[146]. As such simply iterating the process may weaken the security of the protocol and not even be the most efficient way to sign the message.

1. Conflict over Protocol Iteration

The potential issues surrounding iterating a protocol were first raised by Tian-Yin Wang et al.[146] in which a multi-bit signing scheme was proposed that "tagged" the ends of a message. This was later returned to and improved upon[160]. This work gained attention from other research groups who also saw an issue in single bit iteration. Techniques such as ghost imaging[159], quantum but commitment[161] and adaptations of Wang's initial technique[162] have been proposed.

The argument for defining how a protocol handles messages longer than a single bit in length arises as the whole multi-bit message itself isn't encoded anywhere in the signature. Only the value of a single bit is encoded. In a classical signature this is the case and allows for the checking of message integrity. A demonstration of the

the issues of simply iterating a QDS protocol is that of selective attacks. Defining a message as a series of iterated single bits with corresponding private key strings such that the pair received by Bob (M, PK_M) can be broken down as[146]:

$$M = m_1||m_2||...||m_n \tag{32}$$

$$PK = pk_1||pk_2||...||pk_n \tag{33}$$

Where m and pk represent the individual bit components of the multi-bit message and their corresponding private key strings respectively. The corresponding set of quantum signature strings is therefore denoted by:

$$QS = qs_1||qs_2||...||qs_n \tag{34}$$

Where n gives the total bit length of the multi-bit message and || denotes the concatenating of subsequent components. Wang[146] argued that Bob could successfully forge a message by only selectively forwarding certain bit strings. The practical example used in Wang's paper[146] was that M sent by Alice consisted of the bits to represent the phrase "Don't pay Bob $ 100". As this message was produced by iterating a protocol for each bit, each bit of the message m_i would have a corresponding private key string pk_i. Bob is now in possession of a signed message with a set of associated correct signatures. Wang purported that Bob may simply only forward the bits (and corresponding private key strings) that represent for example the message "Pay Bob $ 100"[146160]. As the private key string with each bit will successfully verify when Charlie checks against his stored quantum signature measurements, Bob has successfully forged a message from Alice.

This is not the only attack reported by Wang[146] that could be performed. If Bob has in his possession two separate message and private key sets then, he could separate out the message bits and "stitch" them together in a new order to forge a message. For example if he receives two messages from Alice one stating "Pay Bob $ 100" and another stating "I have $ 200", Bob can choose to only forward the bits in the first part of the first message with the latter part of the second to give "Pay Bob $ 200". As the bit forwarded would have a correct corresponding private key Charlie would verify this message as correct.

These would clearly breach the security of a quantum digital signature protocol. The unbreakable security that is derived from the quantum mechanical effects inherent to the system is let down by an issue in how the protocol is implemented. Despite the minimal further research to support Wang's claim the issue raised by it is still valid. Although the examples mentioned were simplistic this could have serious repercussions in real life applications. For example if Alice had sent Bob a contract then he could choose not to send the bits relating to sections of the contract he did not approve of to Charlie. Furthermore these attacks need not even be performed by Bob, if a malicious third party, Eve, were to intercept (M, PK_M) they could also commit acts of forgery.

It is clear that work is needed in expanding protocols to consider how to sign multi-bit messages safely. Although this work is likely not a priority for many research groups until the rapid secure signing of a single bit is developed it is necessary for the full practical implementation of QDS.

2. *"End Tagging"*

A naive solution to this would be to record the sequence in which the signatures are sent and label the message/private key combinations with the corresponding number[162]. This would not prevent Bob from omitting information at the end of a message however (unless the number of signatures sent exactly matched the number of message bits) or prevent him from "stitching" together messages. The latter it would only make more difficult as he could take the part of one message (say the first five bits of a ten bit message) and 'stitch' it together with another (the last five bits of the second message).

To solve these issues, Wang[146] proposed a protocol to be used as a primitive in an overall multi-bit signing protocol. Any of the single bit signing schemes detailed in this report can be used to sign the individual bits, the "end tagging" process determines how these should be iterated to form a multi-bit message.

In keeping with the notation for a multi-bit message given in section IV F 1 the first step in the proposed method is to create a "sufficiently large"[160] number of private key strings. Each of these are labelled as corresponding to a 0 or 1 message bit (in the same manner as with single bit signing protocols) and also sequentially numbered. The method of encoding as quantum information is independent to the rest of the protocol, as per single bit QDS protocols a copy of each signature is generated for each recipient. The distribution of the set of of quantum signatures S to said recipients is unaffected by this multi-bit expansion. As such any QDS generation and distribution scheme can be used, making this protocol simple to use to extend existing schemes. The recipients then measure each of these to create their own set of signature strings.

Where Wang's proposal tackles the issues of multi-bit encoding is in the encoding of the message. The following

steps are taken to encode M[146]:

1. Encode any bit with the value of 0 as 00.
2. Encode any bit with the value of 1 as 01.
3. Add the codeword 11 to the start and the end of the message.

This results in an output message $\hat{M}$ which has $2n = 4$ elements compared to M's n total elements. To sign a message each bit in the message is assigned a private key string depending on its bit value and on its location in the message (e.g. the first bit will be assigned the first of the private key strings). Alice then sends to Bob the combination of information denoted as $(M, PK_{\hat{M}}, l)$ where the message before encoding is denoted M, the private key strings for each bit in the encoded message denoted $PK_{\hat{M}}$ and l the sequence number of the first key.

Bob then converts M to $\hat{M}$ in the same manner that Alice did. He then applies the authentication method associated with the method of encoding used to each measured string s_i present in the set S. If each string passes authentication then he knows that the message is from Alice and that it has not been tampered with. For secondary verification Bob can forward $(M, PK_{\hat{M}}, l)$ on to Charlie for him to authenticate.

The security of the signing of each individual bit is already well established (see the previous sections in this report) and so does not need to be discussed further here. The end tagging and codewords are what enables this protocol to prevent parts of valid message/signature pairs from being used to create forged messages as described in section IV F 1. In the first attack described Bob forged a message by not forwarding the whole message to Charlie thus, changing the meaning of the message. However, as each message bit would have a correct associated private key Charlie would see this as authentic. The end tagging of each encoded message with the bits 11 prevents this. If Charlie receives a signature set which does not begin and end with two signatures representing 1 then he knows it has been altered. As the bits in the message are encoded to 00 or 01 there is no place the message/signature can be "cut" in order to produce the required 11 needed to mark the beginning and end of a valid message. As the bits are numbered and labelled as to what bit they represent the order of them cannot be changed to achieve this either.

3. *Quantum Temporal Ghost Imaging*

Quantum Temporal Ghost Imaging (QTGI) was developed as a method of speeding up the signing process of a multi-bit message whilst improving resistance to selective attacks. Demonstrated in the initial proposal was the signing of 10 bits of classical information with a single quantum signature[159]. Ghost imaging is the principle in which a single image is created from the output of two separate detectors[163]. In the classical sense two coherent beams are used in the detection process. For QTGI two energy-time entangled photons are used instead.

Key to the principle of QTGI is the three layer encoding within the measurements of the entangled photons[159]. The entire time the measurements were run for is split into a series of "frames". Each frame is then split into a number of "slots" and each slot into four "bins". The size of each is fixed and determined prior to the measurements. The bit size of the message that can be encoded is determined by the chosen slot size.

Alice begins by creating an energy-time entangled photon pair, sending one of the photons to Bob (Charlie) and keeping the other. Alice then passes her photon through an intensity modulator whose period is the same length as a frame. Followed by a low resolution single photon detector. The bit pattern being sent via ghost imaging being the binary pattern of the intensity modulator.

As Alice only has a low resolution SPD she can only detect the frame in which her photon was measured. Bob (Charlie) however possesses a high resolution SPD. Thus, he detects which bin the photons arrive in but does not know which frames contain photons sent by Alice. Thus, without measuring both photons in the entangled pair no one can fully replicate the temporal image of the binary pattern, keeping it secure.

Alice then publicly announces which frames her photons were measured in. Using this Bob (Charlie) can recreate the binary pattern created by Alice's modulator without that explicit information having ever been sent. By then sacrificing part of their records Alice and Bob (Charlie) can then determine the presence of an eavesdropper in a manner similar to that performed in QKD protocols. Bob and Charlie then randomly exchange half of their records to prevent repudiation from Alice.

To sign a message of N bit length Alice then sends to Bob the frame number(s) of the binary string that represents the chosen message[159]. From his own records Bob can then verify the message. Forwarding to Charlie if he requires further verification.

Thus, through the recreation of the temporal image of the intensity modulators pattern via QTGI a multi-bit message can be signed with a single signature[159].

G. Further Extensions

Although the quantum digital signature schemes discussed so far are in theory completely secure, many of them have limitations. For example, assumptions made in theoretical models or even laboratory experiments that are required to simplify the problem are detrimental in practical implementations. Assumptions made regarding channel security allow for focus on just malicious actions by just Alice, Bob or Charlie. In reality it will not be the case that only one of those three are attackers. Other considerations include the practical implementation of such schemes and what side channel attacks can be performed on them. These simplifications are necessary in order to create the theoretical models required but in order for quantum digital signatures to become a viable technology they must be built upon.

1. Insecure Channels

With the exception of the QKD based scheme (see section IV E) each of the schemes so far have made the assumption that quantum information is sent over an authenticated quantum channel[164]. Namely that the information sent is always the same as the information received. This simplifies analysis as it ensures that there is no "Eve" (forger intercepting the information). Whilst the same protections against Bob(Charlie) forging would still apply to Eve, there is no way for the recipients to detect that an interception has occurred until the verification step, at which point the individual who received the legitimate signature would disagree with the one that didn't.

Whilst there are (costly) methods of implementing an authenticated quantum channel[143] this need not always be necessary. First of all we can take a leaf from the book of QKD. As discussed in section IV E Alice and Bob(Charlie) can sacrifice a section of the received QDS to ensure they are sufficiently correlated. If the expected level of error (based on the authentication threshold) is present in this then they can be sure no eavesdropping has occurred. If it is greater than this however, they know Eve is present and can restart the process with a different channel[156].

The QKD based scheme gives us other methods that can be used in order to bypass the issue of insecure channels for any scheme. In particular that of two separate quantum digital signatures being sent from Bob and Charlie to Alice[155]. Assuming Eve does not intercept both the quantum channels then she does not have access to the whole quantum digital signature. This does not allow for the detection of Eve before the verification step but ensures that so long as she does not have access to at least one quantum channel she cannot commit a forgery.

Finally, as proposed with the initial concept for the QKD based QDS scheme is the principle of sending decoy states[164]. A scheme that includes decoy states proceeds with the generation of the two copies of a potential message's quantum signature but before the messages are dispatched they pass through a separate amplitude modulator. This randomly and independently changes the intensity of the signature element pulse to one of three possible values, μ, ν, or 0. Each with their own defined probability distributions. Only the state in which Bob and Charlie both receive μ intensity states is deemed the signal state, the other 6 possible combinations are decoy states. Alice then announces the intensity of each pulse allowing Bob and Charlie to discard any states that were decoys. To any attacker however, there would be no method of determining which states are decoys and which are not. Thus, circumventing the security concerns surrounding the use of insecure quantum channels.

2. Measurement Device Independent

As is the case with many concepts in cryptography, a scheme itself may be secure but in its implementation weaknesses may be found and exploited. These are known as side channel attacks. In classical cryptography and example of this would be analysing the power output of a CPU during encryption in order determine further information about the process used. As such in theoretical models and laboratory tests these are often not considered. Side channel attacks are in fact common in the field of quantum cryptography[143][140]. As there is no way to breach the encoding of the information itself, an attacker must look for exploits elsewhere.

Measurement Device Independent (MDI) schemes bypass these issues by having all quantum communications occur via an untrusted central relay, "Eve"[165], as shown in figure 10. As such none of the other members of the communication actually perform any measurements. They no longer need to be concerned with detector based side channel attacks such as detector blinding as the relay is treated as a "black box".

To do this Alice performs the QKD KGP for QDS technique described in section IV E with Eve. For each state sent, Alice applies random intensity modulation to create a series of decoy states and one signal state (as described in section IV G 1). This is known as a Measurement Device Independent Key Generation Protocol (MDI KGP) and is based off of MDI QKD protocols[158][152].

Eve announces the results of each measurement and their intensity over a public channel. Alice and Bob(Charlie) then communicate over an authenticated classical channel which intensity state was the signal state

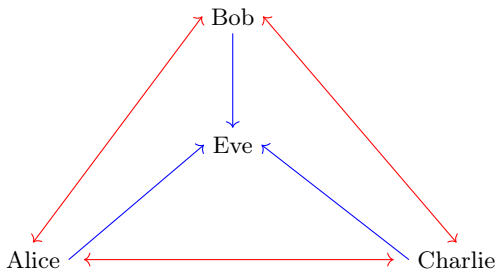

Figure 10: Representation of the communication channels between Alice, Bob, Charlie and the central relay Eve in MDI QDS schemes. Blue arrows represent quantum communication channels and red represent classical channels. Single headed arrows represent one way communication channels, double headed represent two way channels.

as well as which basis was used[165]. Thus generating a signature without direct quantum communication. Bob and Charlie symmetrise their signature strings and the scheme proceeds to the messaging and verification stages.

3. *Expanding to Multiple Parties*

Each scheme that has been discussed has focused on at maximum three parties communicating. A sender of the message (Alice) and two recipients (Bob and Charlie). In a practical communications scenario this obviously will not be practical[152]. In the case of sending a mass message authenticated with a quantum digital signature there will be more than two recipients for the same signature.

However, this raises two concerns. The first is a practical one; in the schemes described above for a practical quantum communication network, each pair of users will require a quantum communication channel. Scaling as $N(N-1)/2$ links for N users[152]. As N increases this number becomes less and less practical to implement. A solution to this issue could be the network architecture discussed in section IV G 2. Rather than each pair of members of the communication network having a quantum communication channel between each other they instead each have one quantum channel with an untrusted central relay. This reduces the scaling of the number of required channels to $N-1$ for an N mode network.

The second issue is a security concern. If Alice sends out the same quantum digital signature to each recipient then, if there are more than two recipients, for N recipients up to a maximum of $N-1$ of them can collude to attempt to commit forgery against the others. Due to the uncertain nature of the measurements of the quantum signature no single recipient will have a fully correct signature. However, by working together multiple malicious parties can work together to improve their chances of a successful forgery. This issue was first addressed in ref[167] where a generic case of a multiparty scheme was first proposed. Ref[168] further expanded upon this concept.

Using the decoy state KGP scheme as a basis (see section IV G 1) they expand this from only having two recipients to a general case of N recipients[167]. To achieve this each recipient generates their own private key for each possible message. They generate a quantum signature for each and distribute to Alice using the process detailed in section IV G 1. Each recipient then randomly chooses half of the bits in their private key and forwards it to each other recipient. Resulting in the final private key for each being $(N-1)L/2$. As such from the viewpoint of Alice each private key is exactly the same. Each recipient as well will never know all of the private keys as each other recipient kept half of theirs. Therefore even if $N-1$ recipients colluded they could not successfully forge a signature from Alice. The proposal also discusses the implementation of a system of security levels to quantify how often a signature can be forwarded and remain safe (as this will affect the authentication threshold)[167]. On top of this is a majority voting protocol for resolving disputes. Thus, fully outlining the protocols required to generalise a QDS protocol to any number of recipients.

4. *Arbitrated Quantum Signatures*

Signature schemes can be split into two different varieties, true and arbitrated (as discussed in chapter 2). The QDS schemes discussed so far in this paper fall within the category of the former. Arbitrated digital signatures however require a third party (of whom does not have to be a trusted party) for the verification of any signature. Due to this key difference arbitrated schemes allow for the resolution of potential conflicts via the impartial arbitrator. Through initial the work of Zeng and Keitel this concept was expanded to creature Arbitrated Quantum Signatures (AQS)[169].

In the generation stage of the initial AQS scheme Alice and Bob each generate a private key that they share with only the arbitrator (dubbed K_A and K_B respectively). The arbitrator then generates a set of GHZ state entangled particles[169]. Keeping one of the set for themselves and sending one to each of Alice and Bob.

In the signing phase Alice entangles the quantum state representation of the message she wishes to sign ($|P\rangle$) with her GHZ particle. She then measures her particle and records the result (M_A). Alice then proceeds to encrypt her message $|P\rangle$ using K_A to create $|R\rangle$. To sign her

Author	Scheme Summary	Distance (km)	Signature Length	Time to sign (s)	Clock rate (Hz)	Security level
Collins et al.[151]	Multiport with USE	0.005	$5.1*10^{13}$		$100*10^{6}$	10^{-2}
Collins et al.[155]	DPS QKD	90	2502		10^{9}	10^{-4}
Donaldson et al.[145]	USE based post-measurement random forwarding	0.5	$1.93*10^{9}$	20		10^{-2}
Croal et al.[157]	Hetrodyne Measurements	1.6	$7*10^{4}$		$2.2*10^{6}$	10^{-2}
Yin et al.[166]	Decoy State	102	$2.5*10^{12}$	33420		10^{-5}
Roberts et al.[152]	MDI-QKD	25	103336	36		
Yin et al.[158]	MDI-QDS (MDI-KGP)		787468			10^{-7}
Yao et al.[159]	Temporal Ghost Imaging		93.9 (signs 10 message bits at a time)	4		10^{-4}

Table V: Summary of the figures of merit of signature schemes referenced in this review. Only includes experimental results and not theoretical estimations of proposed schemes. "Distance" refers to the distance between Alice and Bob or Charlie. "Signature Length" to the bit length of the signature required to sign a single bit message.

message Alice then sends it alongside $|Q_S\rangle = K_A(M_A, |R\rangle)$ to Bob[170].

In the verification stage Bob cannot verify the authenticity of the message himself as he does not possess K_A and so cannot decrypt $|Q_S\rangle$[169]. Instead he encrypts it with his own key, sending this alongside the measurement results of his own GHZ particle and his copy of the message to the arbitrator. This is denoted as $y_B = K_B(M_B, |P\rangle, |Q_S\rangle)$.

Possessing both the private keys the arbitrator can decipher both y_B and Q_S[170]. Using their knowledge of K_A and the copy of $|P\rangle$ received from Bob, the arbitrator creates their own copy of $|R\rangle$, known as $|R'\rangle$. If $|R\rangle = |R'\rangle$ then the signature is successfully verified by the arbitrator. The arbitrator then passes this information, alongside the measurements of their GHZ state (M_T) to Bob.

Bob can then provide further verification by using M_A, M_B and M_T to generate his own copy of $|P\rangle$ to check against the one he received from Alice[170].

This scheme prevents forgery as any information that is sent to another does not reveal either of the private keys[169]. Thus, if Alice and Bob securely communicated these with the arbitrator via QKD, prevent any attacker from successfully forging a signature. As well as this message forgery cannot be successfully achieved as due their is no way of affecting the entangled particles used[169].

AQS schemes have been further developed since their initial inception. The concept has been altered to allow variations such as message recovery[170], the type of entangled states used changed for Bell states[171] and the addition of publicly declared information[172].

H. Concluding Remarks on QDS

The field of quantum digital signatures is a relatively new area of research but in the time it has existed it has developed from a purely theoretical concept into (albeit limited) practical implementation over 100km of fibre[166]. Arguably the most important advance that allowed for feats such as this is that of random forwarding. Whilst simplistic in design this has allowed for the reduction of the quantum technology required in QDS down to simply that used in the already well tested QKD. This paved the way for future schemes which improved upon the basic random forwarding scheme detailed in section IV D. This was achieved either with better hardware or by using it as a primitive for new concepts such as the QKD KGP scheme, which at the moment boasts one of the quickest signing rates over the longest distance.

Quantum digital signatures however, are by no means ready for full commercial use. No current scheme fully solves all of the issues that would allow for QDS to move into broader practical use. For a scheme to be viable for use it must not rely on the assumptions stated in many papers to work. It must be able to work with insecure channels, be immune to side channel attacks and allow for both multiple bits and multiple users securely. The MDI KGP scheme detailed in section IV G 3 (with the recipients sending the signatures to Alice) currently comes

the closest to achieving this. It is secure against forging and repudiation and if adapted to include the multi-bit signing techniques detailed in section IV F would allow for many bit messages to be sent to many recipients. No one has yet, however, attempted to practically implement such a scheme.

V. CONCLUSION

Signatures play a vital role in the security and trustworthiness of communications and, moving forwards, there are valid concerns about the long-term reliability of the digital signature schemes that have been widely adopted. Solutions basing their security on quantum mechanics, rather than complex mathematics are appealing but are far from commercial readiness, and post-quantum digital signatures form an optimal middle ground whilst quantum technologies mature.

VI. ACKNOWLEDGEMENTS

R.J.Y. acknowledges support from the Royal Society through a University Research Fellowship (UF160721). This material was supported by the Air Force Office of Scientific Research under Award No. FA9550-16-1-0276. This work was also supported by grants from The Engineering and Physical Sciences Research Council in the UK (EP/K50421X/1 and EP/ L01548X/1).

VII. DATA AVAILABILITY

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

REFERENCES

[1] Jeremy Norman. The earliest autograph signatures.
[2] W. Diffie and M. Hellman. New directions in cryptography. IEEE Transactions on Information Theory, 22(6):644–654, 1976.
[3] Abelian Foundation. Anonymous digital signatures and their application in cryptocurrency, Jun 2019.
[4] Whitfield Diffie. The first ten years of public-key cryptography. Proceedings of the IEEE, 76(5):560–577, 1988.
[5] Dylan Clarke and Tarvi Martens. E-voting in estonia. Real-World Electronic Voting: Design, Analysis and Deployment, pages 129–141, 2016.
[6] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 21(2):120–126, February 1978.
[7] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory, 31(4):469–472, 1985.
[8] Michael O Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, Massachusetts Inst of Tech Cambridge Lab for Computer Science, 1979.
[9] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Conference on the theory and application of cryptographic techniques, pages 186–194. Springer, 1986.
[10] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing, 17(2):281–308, 1988.
[11] Shafi Goldwasser, Silvio Micali, and Ronald L Rivest. A "paradoxical" solution to the signature problem. In Workshop on the Theory and Application of Cryptographic Techniques, pages 467–467. Springer, 1984.
[12] Steve Wright. Quadratic Residues and Non-Residues. Springer International Publishing, 2016.
[13] Ivan Bjerre Damgård. Collision free hash functions and public key signature schemes. In Workshop on the Theory and Application of of Cryptographic Techniques, pages 203–216. Springer, 1987.
[14] That this seems slightly paradoxical was certainly not lost on them, as seen by the title of their paper.
[15] Wouter Penard and Tim van Werkhoven. On the secure hash algorithm family. Cryptography in Context, pages 1–18, 2008.
[16] Ronald Rivest and S Dusse. The md5 message-digest algorithm, 1992.
[17] Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. Ripemd-160: A strengthened version of ripemd. In International Workshop on Fast Software Encryption, pages 71–82. Springer, 1996.
[18] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures-how to sign with rsa and rabin. In International conference on the theory and applications of cryptographic techniques, pages 399–416. Springer, 1996.
[19] J Jonsson and B Kaliski. Public-key cryptography standards (pkcs)# 1: Rsa cryptography specifications version 2.1. 2003.
[20] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. SIAM Journal on computing, 18(1):186–208, 1989.
[21] Hans Delfs and Helmut Knebl. Introduction to Cryptography: Principles and Applications. Springer, 2015.
[22] CORPORATE NIST. The digital signature standard. Communications of the ACM, 35(7):36–40, 1992.
[23] Rachel Gibson. Elections online: Assessing internet voting in light of the arizona democratic primary. Political Science Quarterly, 116(4):561–583, 2001.
[24] Jason Kitcat and Ian Brown. Observing the english and scottish 2007 e-elections. Parliamentary Affairs, 61(2):380–395, 2008.
[25] Thomas Fujiwara. Voting technology, political responsiveness, and infant health: Evidence from brazil. Econometrica, 83(2):423–464, 2015.
[26] Scott Wolchok, Eric Wustrow, J Alex Halderman, Hari K Prasad, Arun Kankipati, Sai Krishna Sakhamuri, Vasavya Yagati, and Rop Gonggrijp. Security analysis of india's electronic voting machines. In Proceedings of the 17th ACM conference on Computer and communications security, pages 1–14, 2010.
[27] Neal Koblitz. Elliptic curve cryptosystems. Mathematics of computation, 48(177):203–209, 1987.
[28] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ecdsa). International journal of information security, 1(1):36–63, 2001.
[29] S Blake-Wilson, N Bolyard, V Gupta, C Hawk, B Moeller, and Ruhr-uni Bochum. Elliptic curve cryptography (ecc) cipher suites for transport layer security (tls), rfc4492. 2006.
[30] Ralph C Merkle. A certified digital signature. In Conference on the Theory and Application of Cryptology, pages 218–238. Springer, 1989.
[31] Gustavus J Simmons. Contemporary cryptology: The science of information integrity. IEEE press, 1994.
[32] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 400(1818):97–117, 1985.

[33]P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In Proceedings 35th Annual Symposium on Foundations of Computer Science, pages 124–134, 1994.
[34]Fang Xi Lin. Shor's algorithm and the quantum fourier transform. McGill University, 2014.
[35]Carl Pomerance. A tale of two sieves. Biscuits of Number Theory, 85:175, 2008.
[36]P W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Review, 41, 1999.
[37]Richard Jozsa. Quantum factoring, discrete logarithms, and the hidden subgroup problem. Computing in science & engineering, 3(2):34–43, 2001.
[38]Michelangelo Grigni, Leonard Schulman, Monica Vazirani, and Umesh Vazirani. Quantum mechanical algorithms for the nonabelian hidden subgroup problem. In Proceedings of the thirty-third annual ACM symposium on Theory of computing, pages 68–74, 2001.
[39]Chris Lomont. The hidden subgroup problem-review and open problems. arXiv preprint quant-ph/0411037, 2004.
[40]Oded Regev. Quantum computation and lattice problems. CoRR, cs.DS/0304005, 2003.
[41]Lov K Grover. A fast quantum mechanical algorithm for database search. In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pages 212–219, 1996.
[42]Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In Latin American Symposium on Theoretical Informatics, pages 163–169. Springer, 1998.
[43]Daniel J Bernstein, Nadia Heninger, Paul Lou, and Luke Valenta. Post-quantum rsa. In International Workshop on Post-Quantum Cryptography, pages 311–329. Springer, 2017.
[44]Daniel J Bernstein. Introduction to post-quantum cryptography. In Post-quantum cryptography, pages 1–14. Springer, 2009.
[45]Stephen P Jordan and Yi-Kai Liu. Quantum cryptanalysis: Shor, grover, and beyond. IEEE Security & Privacy, 16(5):14–21, 2018.
[46]Enrique Martín-López, Anthony Laing, Thomas Lawson, Roberto Alvarez, Xiao-Qi Zhou, and Jeremy L. O'Brien. Experimental realization of shor's quantum factoring algorithm using qubit recycling. Nature Photonics, 6(11):773–776, Oct 2012.
[47]Zhaokai Li, Nikesh S. Dattani, Xi Chen, Xiaomei Liu, Hengyan Wang, Richard Tanburn, Hongwei Chen, Xinhua Peng, and Jiangfeng Du. High-fidelity adiabatic quantum computation using the intrinsic hamiltonian of a spin system: Application to the experimental factorization of 291311, 2017.
[48]Gorjan Alagic, Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, 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.
[49]National Institute of Standards and Technology. 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`, 2016.
[50]Joan Daemen and Vincent Rijmen. Aes proposal: Rijndael. 1999.
[51]National Institute of Standards and Technology. Round 3 submissions - post-quantum cryptography @ONLINE. `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`, 2020.
[52]Jintai Ding, Ming-Shing Chen, Albrecht Petzoldt, Dieter Schmidt, and Bo-Yin Yang. Rainbow - nist pqc submission @ONLINE. `https://csrc.nist.gov/CSRC/media/Presentations/Rainbow/images-media/Rainbow-April2018.pdf`, 2020.
[53]Vadim Lubashevsky, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Peter Schwabe, Gregor Siler, and Damien Stehle. Dilithium - nist pqc submission @ONLINE. `https://pq-crystals.org/dilithium/index.shtml`, 2020.
[54]Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lubashevsky, Thomas Pronin, Thomas Ricosset, Gregor Siler, William Whyte, and Zhenfei Zhang. Falcon - nist pqc submission @ONLINE. `https://falcon-sign.info/`, 2020.
[55]Simona Samardjiska, Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, and Peter Schwabe. Mqdss - nist pqc submission @ONLINE. `http://mqdss.org/index.html`, 2020.
[56]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 - nist pqc submission @ONLINE. `https://www.microsoft.com/en-us/research/project/picnic/`, 2020.
[57]Andreas Hülsing, Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Pano Kampanakis, Stefan Kolbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, and Jean-Phillippe Aumasson. Sphincs$^+$ - nist pqc submission @ONLINE. `https://sphincs.org/index.html`, 2020.
[58]Hideki Imai and Tsutomu Matsumoto. Algebraic methods for constructing asymmetric cryptosystems. In International Conference on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes, pages 108–119. Springer, 1985.
[59]Jacques Patarin. Cryptanalysis of the matsumoto and imai public key scheme of eurocrypt'88. In Annual International Cryptology Conference, pages 248–261. Springer, 1995.
[60]Christopher Wolf. Introduction to multivariate quadratic public key systems and their applications. Proceedings of YACC, pages 44–55, 2006.
[61]Harry R Lewis. Michael r. πgarey and david s. johnson. computers and intractability. a guide to the theory of np-completeness. wh freeman and company, san francisco1979, x+ 338 pp. The Journal of Symbolic Logic, 48(2):498–500, 1983.
[62]Takanori Yasuda, Xavier Dahan, Yun-Ju Huang, Tsuyoshi Takagi, and Kouichi Sakurai. Mq challenge: Hardness evaluation of solving multivariate quadratic problems. IACR Cryptology ePrint Archive, 2015:275, 2015.
[63]Jacques Patarin. The oil and vinegar signature scheme. In Dagstuhl Workshop on Cryptography September, 1997, 1997.
[64]Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced oil and vinegar signature schemes. In International Conference on the Theory and Applications of Cryptographic Techniques, pages 206–222. Springer, 1999.
[65]Christopher Wolf. " Hidden Field Equations"(HFE)-Variations and Attacks. PhD thesis, Verlag nicht ermittelbar, 2002.
[66]Jean-Charles Faugere, Françoise Levy-Dit-Vehel, and Ludovic Perret. Cryptanalysis of minrank. In Annual International Cryptology Conference, pages 280–296. Springer, 2008.
[67]Vivien Dubois, Louis Granboulan, and Jacques Stern. Cryptanalysis of hfe with internal perturbation. In International Workshop on Public Key Cryptography, pages 249–265. Springer, 2007.
[68]Olivier Billet and Jintai Ding. Overview of cryptanalysis techniques in multivariate public key cryptography. In Gröbner Bases, Coding, and Cryptography, pages 263–283. Springer, 2009.
[69]Miklós Ajtai. Generating hard instances of lattice problems. In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pages 99–108, 1996.
[70]Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. Ntru: A ring-based public key cryptosystem. In International Algorithmic Number Theory Symposium, pages 267–288. Springer, 1998.

[71]Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM (JACM), 56(6):1–40, 2009.
[72]Chris Peikert et al. A decade of lattice cryptography. Foundations and Trends® in Theoretical Computer Science, 10(4):283–424, 2016.
[73]Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 1–23. Springer, 2010.
[74]Charles Grover. LWE Over Cyclic Algebras: A Novel Structure for Lattice Cryptography. PhD thesis, Imperial College London, 2020. Contact authors for copy.
[75]Philip Klein. Finding the closest lattice vector when it's unusually close. In Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms, pages 937–941, 2000.
[76]James Howe, Ayesha Khalid, Ciara Rafferty, Francesco Regazzoni, and Máire O'Neill. On practical discrete gaussian samplers for lattice-based cryptography. IEEE Transactions on Computers, 67(3):322–334, 2016.
[77]Thomas Prest. Gaussian sampling in lattice-based cryptography. PhD thesis, 2015.
[78]Miklós Ajtai. The shortest vector problem in l2 is np-hard for randomized reductions. In Proceedings of the thirtieth annual ACM symposium on Theory of computing, pages 10–19, 1998.
[79]Dorit Aharonov and Oded Regev. Lattice problems in np ∩ conp. Journal of the ACM (JACM), 52(5):749–765, 2005.
[80]Martin R Albrecht and Amit Deo. Large modulus ring-lwe ≥ module-lwe. In International Conference on the Theory and Application of Cryptology and Information Security, pages 267–296. Springer, 2017.
[81]Miruna Roşca, Amin Sakzad, Damien Stehlé, and Ron Steinfeld. Middle-product learning with errors. In Annual International Cryptology Conference, pages 283–297. Springer, 2017.
[82]Charles Grover, Cong Ling, and Roope Vehkalahti. Non-commutative ring learning with errors from cyclic algebras.
[83]Chris Peikert. How (not) to instantiate ring-lwe. In International Conference on Security and Cryptography for Networks, pages 411–430. Springer, 2016.
[84]Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. Designs, Codes and Cryptography, 75(3):565–599, 2015.
[85]Adeline Roux-Langlois. Lattice-Based Cryptography-Security Foundations and Constructions. PhD thesis, 2014.
[86]Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Proceedings of the fortieth annual ACM symposium on Theory of computing, pages 197–206, 2008.
[87]Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In Annual International Cryptology Conference, pages 112–131. Springer, 1997.
[88]Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H Silverman, and William Whyte. Ntrusign: Digital signatures using the ntru lattice. In Cryptographers' Track at the RSA Conference, pages 122–140. Springer, 2003.
[89]Phong Nguyen. Cryptanalysis of the goldreich-goldwasser-halevi cryptosystem from crypto'97. In Annual International Cryptology Conference, pages 288–304. Springer, 1999.
[90]Craig Gentry and Mike Szydlo. Cryptanalysis of the revised ntru signature scheme. In International Conference on the Theory and Applications of Cryptographic Techniques, pages 299–320. Springer, 2002.
[91]Léo Ducas and Phong Q Nguyen. Learning a zonotope and more: Cryptanalysis of ntrusign countermeasures. In International Conference on the Theory and Application of Cryptology and Information Security, pages 433–450. Springer, 2012.
[92]László Babai. On lovász'lattice reduction and the nearest lattice point problem. Combinatorica, 6(1):1–13, 1986.
[93]Shuichi Katsumata, Shota Yamada, and Takashi Yamakawa. Tighter security proofs for gpv-ibe in the quantum random oracle model. In International Conference on the Theory and Application of Cryptology and Information Security, pages 253–282. Springer, 2018.
[94]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.
[95]Léo Ducas and Thomas Prest. Fast fourier orthogonalization. In Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, pages 191–198, 2016.
[96]Shi Bai and Steven D Galbraith. An improved compression technique for signatures based on learning with errors. In Cryptographers' Track at the RSA Conference, pages 28–47. Springer, 2014.
[97]Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In International Conference on the Theory and Application of Cryptology and Information Security, pages 598–616. Springer, 2009.
[98]Vadim Lyubashevsky and Daniele Micciancio. Asymptotically efficient lattice-based digital signatures. In Theory of Cryptography Conference, pages 37–54. Springer, 2008.
[99]Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In Annual Cryptology Conference, pages 40–56. Springer, 2013.
[100]Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In International Workshop on Cryptographic Hardware and Embedded Systems, pages 530–547. Springer, 2012.
[101]Vadim Lyubashevsky. Lattice signatures without trapdoors. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 738–755. Springer, 2012.
[102]Claus Peter Schnorr. Progress on lll and lattice reduction. In The LLL Algorithm, pages 145–178. Springer, 2009.
[103]Shanxiang Lyu, Christian Porter, and Cong Ling. Lattice reduction over imaginary quadratic fields with an application to compute-and-forward. arXiv preprint arXiv:1806.03113, 2018.
[104]Yoshinori Aono, Phong Q Nguyen, and Yixin Shen. Quantum lattice enumeration and tweaking discrete pruning. In International Conference on the Theory and Application of Cryptology and Information Security, pages 405–434. Springer, 2018.
[105]Thijs Laarhoven. Sieving for shortest vectors in lattices using angular locality-sensitive hashing. In Annual Cryptology Conference, pages 3–22. Springer, 2015.
[106]Thijs Laarhoven, Michele Mosca, and Joop Van De Pol. Finding shortest lattice vectors faster using quantum search. Designs, Codes and Cryptography, 77(2-3):375–400, 2015.
[107]David Joseph, Alexandros Ghionis, Cong Ling, and Florian Mintert. Not-so-adiabatic quantum computation for the shortest vector problem. Phys. Rev. Research, 2:013361, Mar 2020.
[108]David Joseph, Adam Callison, Cong Ling, and Florian Mintert. Two quantum ising algorithms for the shortest vector problem: one for now and one for later, 2020.
[109]Miklós Ajtai, Ravi Kumar, and Dandapani Sivakumar. Sampling short lattice vectors and the closest lattice vector problem. In Proceedings 17th IEEE Annual Conference on Computational Complexity, pages 53–57. IEEE, 2002.
[110]Divesh Aggarwal, Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. Solving the shortest vector problem in 2n time using discrete gaussian sampling. In Proceedings of the forty-seventh annual ACM symposium on Theory of computing, pages 733–742, 2015.
[111]Jingwen Suo, Licheng Wang, Sijia Yang, Wenjie Zheng, and Jiankang Zhang. Quantum algorithms for typical hard problems: a perspective of cryptanalysis. Quantum Information

Processing, 19:178, 2020.
[112] Leo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehle. Dilithium design document @ONLINE. https://pq-crystals.org/dilithium/data/dilithium-specification-round2.pdf, 2019.
[113] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon design document @ONLINE. https://falcon-sign.info/falcon.pdf, 2019.
[114] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In Proceedings of the 2017 acm sigsac conference on computer and communications security, pages 1825–1842, 2017.
[115] Dominique Unruh. Quantum proofs of knowledge. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 135–152. Springer, 2012.
[116] Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 755–784. Springer, 2015.
[117] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. Zkboo: Faster zero-knowledge for boolean circuits. In 25th {usenix} security symposium ({usenix} security 16), pages 1069–1083, 2016.
[118] Martin R Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for mpc and fhe. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 430–454. Springer, 2015.
[119] Daniel J Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. The sphincs+ signature framework. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, pages 2129–2146, 2019.
[120] Daniel J Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. Sphincs: practical stateless hash-based signatures. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 368–397. Springer, 2015.
[121] Oded Goldreich. Two remarks concerning the goldwasser-micali-rivest signature scheme. In Conference on the Theory and Application of Cryptographic Techniques, pages 104–110. Springer, 1986.
[122] Itai Dinur and Niv Nadler. Multi-target attacks on the picnic signature scheme and related protocols. Cryptology ePrint Archive, Report 2018/1212, 2018. https://eprint.iacr.org/2018/1212.
[123] Christian Rechberger, Hadi Soleimany, and Tyge Tiessen. Cryptanalysis of low-data instances of full lowmcv2. Cryptology ePrint Archive, Report 2018/859, 2018. https://eprint.iacr.org/2018/859.
[124] 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 design document @ONLINE. https://github.com/Microsoft/Picnic/tree/master/spec, 2020.
[125] Laurent Castelnovi, Ange Martinelli, and Thomas Prest. Grafting trees: a fault attack against the sphincs framework. In International Conference on Post-Quantum Cryptography, pages 165–184. Springer, 2018.
[126] Aymeric Genêt, Matthias J Kannwischer, Hervé Pelletier, and Andrew McLauchlan. Practical fault injection attacks on sphincs. IACR Cryptology ePrint Archive, 2018:674, 2018.
[127] Jean-Philippe Aumasson, Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Andreas Hülsing, Panos Kampanakis, Stefan Köbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Rijneveld Joost, and Peter Schwabe. Sphincs$^+$ design document @ONLINE. https://sphincs.org/data/sphincs+-round2-specification.pdf, 2019.
[128] Jonathan Katz and Yehuda Lindell. Introduction to modern cryptography. CRC press, 2014.
[129] Junfeng Fan, Xu Guo, Elke De Mulder, Patrick Schaumont, Bart Preneel, and Ingrid Verbauwhede. State-of-the-art of secure ecc implementations: a survey on known side-channel attacks and countermeasures. In 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pages 76–87. IEEE, 2010.
[130] A Tawalbeh Lo'ai, Turki F Somani, and Hilal Houssain. Towards secure communications: Review of side channel attacks and countermeasures on ecc. In 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST), pages 87–91. IEEE, 2016.
[131] Yasufumi Hashimoto, Tsuyoshi Takagi, and Kouichi Sakurai. General fault attacks on multivariate public key cryptosystems. IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, 96(1):196–205, 2013.
[132] Haibo Yi and Zhe Nie. Side-channel security analysis of uov signature for cloud-based internet of things. Future Generation Computer Systems, 86:704–708, 2018.
[133] Juliane Krämer and Mirjam Loiero. Fault attacks on uov and rainbow. In International Workshop on Constructive Side-Channel Analysis and Secure Design, pages 193–214. Springer, 2019.
[134] Jintai Ding, Zheng Zhang, Joshua Deaton, Kurt Schmidt, and F Vishakha. New attacks on lifted unbalanced oil vinegar. In The 2nd NIST PQC Standardization Conference, 2019.
[135] James Howe Anupam Chattopadhyay Prasanna Ravi, Mahabir Prasad Jhanwar and Shivam Bhasin. Side-channel assisted existential forgery attack on dilithium - a nist pqc candidate. Cryptology ePrint Archive, Report 2018/821, 2018. https://eprint.iacr.org/2018/821.
[136] Vincent Migliore, Benoit Gérard, Mehdi Tibouchi, and Pierre-Alain Fouque. Masking dilithium: Efficient implementation and side-channel evaluation. Cryptology ePrint Archive, Report 2019/394, 2019. https://eprint.iacr.org/2019/394.
[137] Sarah McCarthy, James Howe, Neil Smyth, Séamus Brannigan, and Máire O'Neill. Bearz attack falcon: implementation attacks with countermeasures on the falcon signature scheme. SECRYPT, pages 61–71, 2019.
[138] A. Casanova, J.-C. Faugere, G. Macario-Rat, J. Patarin, L. Perret, and J. Ryckeghem. Gemss - nist pqc submission @ONLINE. https://www-polsys.lip6.fr/Links/NIST/GeMSS.html, 2020.
[139] Matt Braithwaite. Experimenting with post-quantum cryptography @ONLINE. https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html, 2019.
[140] Feihu Xu, Xiongfeng Ma, Qiang Zhang, et al. Quantum cryptography with realistic devices. arXiv, 2019.
[141] Thomas McGrath, Ibrahim E Bagci, Zhiming M Wang, et al. A puf taxonomy. Applied Physics Reviews, 6, 2019.
[142] Daniel Gottesman and Issac L Chuang. Quantum digital signatures. arXiv, 2001.
[143] S Pirandola, U L Andersen, L Banchi, et al. Advances in quantum cryptography. arXiv, 2019.
[144] H Singh, D L Gupta, and A K Singh. Quantum key distribution protocols: A review. IOSR-JCE, 2014.
[145] Ross J Donaldson, Robert J Collins, Klaudia Kleczkowska, et al. Experimental demonstration of kilometer-range quantum digital signatures. Physical Review, 93, 2016.
[146] Tian-Yin Wang, Xiao-Qiu Cai, Yan-Li Ren, et al. Security of quantum digital signatures for classical messages. Scientific Reports, 5, 2015.
[147] Erika Andersson, Marcos Curty, and Igor Jex. Experimentally realizable quantum comparison of coherent states and its appli-

cations. Physical Review A, 74, 2006.
[148]A Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. Probl. Peredachi Inf, 9, 1973.
[149]Vedran Dunjko, Petros Wallden, and Erika Andersson. Quantum digital signatures without quantum memory. Physical Review Letters, 112, 2014.
[150]M Bouillard, G Boucher, J Ferrer Ortas, et al. Quantum storage of single-photon and two-photon fock states with an all-optical quantum memory. Phys Rev Letts, 122, 2019.
[151]Robert J Collins, Ross J Donaldson, Vedran Dunjko, et al. Realization of quantum digital signatures without the requirement of quantum memory. Phys Rev Letts, 113, 2014.
[152]G L Roberts, M Lucamarini, Z L Yuan, et al. Experimental measurement-device-independent quantum digital signatures. Nature Communications, 8, 2017.
[153]Patrick J Clarke, Robert J Collins, Vedran Dunjko, et al. Experimental demonstration of quantum digital signatures using phase-encoded coherent states of light. nature Communcications, 2, 2012.
[154]Petros Wallden and Vedran Dunjko. Quantum digital signatures with quantum-key-distribution components. Phys Rev A, 2015.
[155]Robert J Collins, Ryan Amiri, Mikio Fujiwara, et al. Experimental transmission of quantum digital signatures over 90 km of installed optical fiber using a differential phase shift quantum key distribution system. Optics Letters, 2016.
[156]Ryan Amiri, Petros Wallden, Adrian Kent, et al. Secure quantum signatures using insecure quantum channels. Phys Rev A, 2016.
[157]Callum Croal, Christian Peuntinger, Bettina Heim, et al. Free-space quantum signatures using heterodyne measurements. Phys Rev Lett, 2016.
[158]Hua-Lei Yin, Wei-Long Wang, Yan-Lin Tang, et al. Experimental measurement-device-independent quantum digital signatures over a metropolitan network. Phys Rev A, 2017.
[159]Xin Yao, Xu Liu, Rong Xue, et al. Multi-bit quantum digital signature based on quantum temporal ghost imaging. arXiv, 2019.
[160]Tian-Yin Wang, Jian-Feng Ma, and Xiao-Qiu Cai. The post-processing of quantum digital signatures. Quantum Inf Process, 2017.
[161]Ming-Qiang Wang, Xue Wang, and Tao Zhan. An efficient quantum digital signature for classical messages. Quantum Information Processing, 2018.
[162]Hao Zhang, Xue-Bi An, Chun-Hui Zhang, et al. High-efficiency quantum digital signature scheme for signing longmessages. Quantum Information Processing, 2019.
[163]Miles Padgett and Robert Boyd. An introduction to ghost imaging:quantum and classical. Phil.Trans.R.Soc.A, 2017.
[164]Hua-Lei Yin, Yao Fu, and Zeng-Bing Chen. Practical quantum digital signature. Phys Rev A, 2016.
[165]Ittoop Vergheese Puthoor, Ryan Amiri, Petros Wallden, et al. Measurement-device-independent quantum digital signatures. Phys Rev A, 2016.
[166]Hua-Lei Yin, Yao Fu, Qi-Jie Tang, et al. Experimental quantum digital signature over 102 km. Phys Rev A, 2017.
[167]Juan Miguel Arrazola, Petros Wallden, and Erika Andersson. Multiparty quantum signature schemes. Quantum Info. Comput., 2016.
[168]Mustafa Sahin and Ihsan Yilmaz. Multi-partied quantum digital signature scheme without assumptions on quantum channel security. Journal of Physics: Conference Series, 2016.
[169]Guihua Zeng and Christopher Keitel. Arbitrated quantum-signature scheme. Phys Rev A, 2002.
[170]Hwayen Lee, Changho Hong, Hyunsang Kim, Jongin Lim, and Hyung Jin Yang. Arbitrated quantum signature scheme with message recovery. Phys Letts A, 2004.
[171]Qin Li, W H Chan, and Dong-Yang Long. Arbitrated quantum signature scheme using bell states. Phys Rev A, 2009.
[172]Xiangfu Zou and Daowen Qiu. Security analysis and improvements of arbitrated quantum signature schemes. Phys Rev A, 2010.