Quantum Digital Signatures

Dominic Moylett*
Quantum Engineering Centre for Doctoral Training
University of Bristol

(Dated: May 22, 2016)

While classical cryptography has relied on the assumed existence of one-way functions for proofs of security, quantum cryptography has given us information theoretically secure secret sharing. This has been achieved via quantum key distribution, utilising the uncertainty principle to limit what an eavesdropper can discover about a shared private key. Another application where we can have provably secure primitives is that of quantum digital signatures, which are used for authenticating that a message did come from a specific user. In this essay, we summarise the theoretical and experimental achievements in quantum digital signatures, and discuss their current limitations.

I. INTRODUCTION

Cryptography is the study of secure communications between multiple parties, often called Alice and Bob. In classical cryptography, this security is guaranteed through the assumed hardness of certain problems. The RSA encryption scheme [1], for example, is secure under the assumption that the problem of factoring large integers is computationally hard. Unfortunately, this assumption has no strict guarantees behind it. And even worse, some of these problems, such as factoring, have been shown to be computationally insecure in both theory [2] and experimental work [3, 4] using quantum computers.

Thankfully, while quantum computers have the possibility of breaking secrecy, they also have the potential to fix it. Quantum Key Distribution (QKD) protocols such as BB84 [5] use the uncertainty principle to securely establish a shared key with no assumptions. Once they have a shared key, they can use a one-time pad to send messages with perfect secrecy. This means that it is impossible for an eavesdropper, often called Eve, to learn any information about what Alice and Bob said to one another.

You might think that this solves all the problems of cryptography. But alas, this is not the case, as secrecy is only one aspect of what we want in cryptography. Even if Alice and Bob have guaranteed secrecy, that still doesn't stop Eve from sending a message of random gibberish to Bob while pretending to be Alice. And unless extra precautions are taken, Bob cannot tell whether the random message is forged by an attacker or what Alice wanted to send in the first place. This has also been demonstrated experimentally [6] and even now has industry involvement, with companies such as ID Quantique [7] offering commercial products for QKD.

Those extra precautions are to sign a message using digital signatures. Like how someone signs the end of a letter to indicate that they wrote it, a digital signature is an extra piece of data in a message to indicate who wrote it. And like classical encryption schemes, classical digital signatures rely on the hardness of certain problems such as the Secure Hashing Algorithm (SHA).

Based on the improvements QKD has made in terms of secrecy, one might ask if quantum physics can also benefit digital signatures. This is the subject that this essay will explore. SEC. II will provide mathematical preliminaries before defining a digital signature in SEC. III. SEC. IV describes the classical digital signature that most quantum digital signatures are based on, which are then describe theoretically in SEC. V and experimentally in SEC. VI. Finally, we conclude with some points of discussion in SEC. VII.

II. PRELIMINARIES

Definition II.1. *A function* $f : \mathbb{N} \to \mathbb{R}$ *is negligible iff* $\forall\ c > 0\ \exists\ n_0 \in \mathbb{N}$ *such that* $\forall\ n \geq n_0, |f(n)| < \frac{1}{n^c}$.

The most common example of negligible functions are functions of the form $2^{-p(n)}$ for some positive polynomial p.

III. DEFINITION OF DIGITAL SIGNATURES

A digital signature scheme consists of the following stages:

Key Generation: Alice generates a signing key sk and a verification key vk. The signing key is kept to herself, while the verification key is shared with other parties.

Message Signing: Alice takes a message m and her signing key sk and produces a signature for that message s_m.

Message Verification: Another party, Bob, takes a message m, Alice's verification key vk and a signature s_m and accepts if they think the message was sent and signed by Alice, otherwise they reject.

* dominic.moylett@bristol.ac.uk

Another definition of verification used in, for example, [8] has three different options for Bob. In this definition Bob can output:

1-ACC: He thinks the message-signature pair is valid and other parties will also find the message to be valid.

0-ACC: He thinks the message-signature pair is valid and other parties will not find the message to be valid.

REJ: He thinks the message-signature pair is invalid.

For security, we care about two properties with digital signatures:

Repudiation: The probability that Alice is able to convince Bob that a message-signature pair is valid and convince another party Charlie that the same pair is invalid is negligible in terms of the security parameter (often the length of the signature).

Forgery: The probability that Bob can in polynomial time create a valid message-signature pair he has not previously seen sent by Alice is negligible in terms of the security parameter.

IV. LAMPORT'S ONE-TIME DIGITAL SIGNATURE

The original quantum digital signature was inspired from a classical family of digital signatures. This family of signatures were first published by Lamport[9], and are reliant on the existence of one-way functions, defined below.

Definition IV.1. *Let $\mathcal{X}, \mathcal{Y}$ be arbitrary sets. A function $f : \mathcal{X} \to \mathcal{Y}$ is one-way iff f can be computed in polynomial time, but for any polynomial-time randomised adversary $f^{-1} : \mathcal{Y} \to \mathcal{X}$ and uniformly selected $x \in \mathcal{X}, \Pr[f(f^{-1}(x)) = f(x)]$ is negligible in terms of the security parameter (often the length of x in bits).*

For a one-way function f, then Lamport's one-time digital signature scheme is defined as follows for Alice signing an m-bit message:

Key Generation: Alice generates m pairs of uniformly selected random integers $\{k_0^i, k_1^i\}$. These will act as Alice's signing key, while the verification key will be the pairs $\{f(k_0^i), f(k_1^i)\}$ for $i \in \{0, ..., m-1\}$.

Message Signing: For each bit m_i of Alice's message, she sends $k_{m_i}^i$ as her signature for that bit.

Message Verification: Given Alice's verification key $\{v_0^i, v_1^i\}$ and her signature $k_{m_i}^i$, her message m_i can be verified by accepting if $v_{m_i}^i = f(k_{m_i}^i)$ and rejecting otherwise.

Because this signature scheme is classical and deterministic, it is trivial for any verifying parties to share their copies of the message and verification keys and thus impossible for Alice to repudiate the message. As for forgery, if it is possible for an adversary to forge a message given only one message from Alice then it is possible to invert f, thus f is not a one-way function.

Note that this signature scheme can only be used for verifying a single m-bit message. This is because if an attacker has signatures for two different messages M, M' then they can construct the valid signature for the new message $m_0 m_1 ... m_i' ... m_{m-1}$, where $m_i \neq m_i'$.

V. QUANTUM DIGITAL SIGNATURES

A. Gottesman and Chuang

The first quantum digital signature was devised in 2001 by Daniel Gottesman and Isaac Chuang [8]. The protocol used the same principles as Lamport's one-time signature scheme, but utilised quantum one-way functions, which map classical L-bit strings k to n-qubit quantum states $|f_k\rangle$. The full protocol is described below, where f is the quantum one-way function, c_1, c_2 are security thresholds such that $0 \leq c_1 < c_2 < 1$, M is the security parameter, and $T < L/n$ is the number of copies of each key:

Key Generation: To sign a single bit, Alice generates pairs of L-bit strings $\{k_0^i, k_1^i\}, i \in \{0, ..., M-1\}$ uniformly at random to act as her private key. Her public key is the pairs of quantum states $\{|f_{k_0^i}\rangle, |f_{k_1^i}\rangle\}, i \in \{0, ..., M-1\}$.

Message Signing: To sign a bit b, Alice sends the tuple $(b, k_b^0, k_b^1, ..., k_b^{M-1})$.

Message Verification: To verify a signed message $(b, k_b^0, k_b^1, ..., k_b^{M-1})$, a recipient can use f and SWAP tests [10] to determine if $|f_{k_b^i}\rangle$ matches the corresponding public key. The number of times a recipient finds that the SWAP test fails is noted as s. If $s \leq c_1 M$ then the recipient accepts the message (**1-ACC**), and if $s \geq c_2 M$ then the recipient rejects the message (**REJ**). In the case where $c_1 < s \leq c_2$, the recipient notes that the message is valid but might not be transferable (**0-ACC**).

Security against forgery is possible due to Holevo's theorem [11], which states that even if an eavesdropper managed to acquire all T copies of the public key, they could only extract at most Tn bits of data, and thus their probability of correctly guessing a single L-bit string of the signing key is $2^{-(L-Tn)}$. If the distance between two public keys $|\langle f_k | f_{k'} \rangle| \leq \delta$ for $k \neq k'$, then the probability of an attacker managing to make a recipient accept a signature they weren't able to correctly guess is at most δ^2. Thus by choosing c_2 smaller than the probability of a guessing adversary being accepted

$(1-\delta^2)(1-2^{1-(L-Tn)})$, we can ensure that each recipient rejects a message with high probability.

While protection against forgery is simple enough, it is harder to protect against repudiation. This is because in a two-recipient case, Alice can send Bob and Charlie different states. Due to the no-cloning theorem [12], there is no perfect way for Bob and Charlie to check that their keys are the same. Gottesman and Chuang's solution is to introduce another step to the protocol, called Key Distribution, which uses a distributed SWAP test to check that the keys sent to Bob and Charlie match:

Key Distribution: Alice sends two copies of each key to Bob and Charlie. Bob and Charlie pick random indices $i \in \{0, ..., M-1\}$ to perform a distributed SWAP on. Some of these indices they will perform SWAP tests on their own copies of the key to ensure that those match, and others Charlie will send a copy of their key to Bob so that Bob can perform a SWAP test to ensure that Bob's keys match Charlie's. If any tests fail then the protocol is aborted, otherwise they discard the test keys and continue the protocol with the remaining keys.

With this step in place, the probability of Alice being able to successfully repudiate is the probability that all distributed SWAP tests pass yet $|s_B - s_C| > (c_2 - c_1)M$, where s_B and s_C are Bob and Charlie's incorrect counts, respectively. This probability can be exponentially small, depending on how large the gap between c_1 and c_2 is.

While Lamport's signature scheme assumes the existence of classical one-way functions, no such assumption is required here. This is because quantum one-way functions can be devised, by mapping the L-bit strings to n-qubit states such that the states are nearly orthogonal, meaning that $|\langle f_k|f_k\rangle| \leq \delta$. It was shown by Buhrman et al. [10] that if $\delta \approx 0.9$ then $L = 2^n$. Thus using fingerprint states as our one-way function makes it possible to create an exponentially small probability of forgery.

However, this protocol has a number of disadvantages as well:

1. Verification keys cannot simply be copied due to the no-cloning theorem [12], so all of the keys must come from Alice. This requires authenticated quantum channels from Alice to each recipient, to ensure a forger doesn't simply send their own keys. But for generic quantum states, this is impossible [13], thus offering us an even harder challenge than the classical authentication we're trying to solve.

2. The recipients need to store the keys indefinitely until Alice uses them to sign a message. This requires long-time quantum memory, which is currently impractical.

3. Generating and sharing a quantum one-way function, while possible, is a complex task.

4. The distributed SWAP tests are not efficient.

Over the rest of this section, we will summarise how these problems have been overcome in the years following this result.

B. Initial Improvements

Improvements were initially made to digital signatures by Dunjko, Wallden and Andersson [14], who proposed a system which removed the need for quantum memory, the quantum SWAP test and the cost of sending quantum one-way functions by encoding the keys in coherent states. Their protocol for three parties was devised as follows, with the constraint that $0 \leq s_b < s_c < 1$:

Key Generation: Alice generates two L-bit keys k^0, k^1 at random for some security parameter L, which act as Alice's signing keys. Alice then generates states of the form $|(-1)^{k_l^b}\alpha\rangle$ for $l \in \{0, ..., L-1\}$ and $b \in \{0, 1\}$, where α is a positive real number known between all parties and k_l^b is the l-th bit of k^b. These quantum states will act as the verification key.

Key Distribution: For two recipients, Bob and Charlie, Alice generates a copy of the above state for each participant. Bob and Charlie run their states together through a multiport. For states which come out of the signal port, they then use Unambiguous State Discrimination (USD) [15] to distinguish between the states $\{|-\alpha\rangle, |\alpha\rangle\}$, and for each unambiguous state $|(-1)^{k_l^b}\alpha\rangle$ noting the value of k_l^b, the index l and whether $b = 0$ or 1.

Message Signing: To sign a message b, Alice sends the message (b, k^b) to Bob.

Message Verification: To verify a message-signature pair (b, k^b), Bob (resp. Charlie) checks the bits of k^b sent to him with the bits that he was able to unambiguously determine from the distribution stage. If the number of mismatches is at most $s_b p_{USD} L$ (resp. $s_c p_{USD} L$), where s_b (resp. s_c) are acceptance thresholds and p_{USD} is the USD success probability, then Bob (resp. Charlie) accepts the pair.

The use of the multiport is to defend against repudiation; even if Alice sends $|\alpha\rangle$ to Bob and $|\beta\rangle$ to Charlie such that $\alpha \neq \beta$, only using photons that come out the signal outputs of the multiport ensures that both recipients finish with the state $|(\alpha+\beta)/2\rangle$. Because they have the same reduced state, on average they will see the same number of mismatches. Alice's best strategy has probability $(s_b + s_c)p_{USD}/2$ of making Bob and Charlie disagree on a single bit of the key, which leads to her probability of successful repudiation being negligible in terms of the difference between Charlie and Bob's thresholds, p_{USD} and L.

In order for Bob – or alternatively Charlie – to successfully forge a message, Bob must guess the bits that he was not able to unambiguously get from Alice, yet Charlie was able to. The probability of this is negligible in terms of s_c, p_{USD} and L. Bob's other option is to send wrong states through the multiport, which Charlie can also defend against by aborting if too many photons are seen through his null output [16].

C. Quantum Digital Signatures using QKD

Wallden et al. [17] developed in 2015 an improved signature scheme that no longer required the use of a multiport for Bob and Charlie. This scheme utilises quantum state elimination [18] to build trust in Alice's signature. The protocol is explained below, using $|0\rangle, |1\rangle, |+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}, |-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ are the BB84 [5] states, and $0 \leq s_b < s_c < 1$:

Key Generation: For a future one-bit message, Alice generates pairs $(k_l^0, k_l^1), 0 \leq l < L$ where L is a security parameter and $k \in \{0, 1, +, -\}$ is chosen at random. She then generates two copies of the states $\bigotimes_{l=0}^{L-1} |k_l^0\rangle$ and $\bigotimes_{l=0}^{L-1} |k_l^1\rangle$ to act as her verification keys.

Key Distribution: Alice sends copies of the verification keys to Bob and Charlie. Bob (resp. Charlie) picks between $L/2 - r$ and $L/2 + r$ indices $0 \leq l < L$ and bits $b \in \{0, 1\}$ for some r and sends $(l, b, |k_l^b\rangle)$ to Charlie (resp. Bob), aborting if he receives a number of tuples outside of that limit. Bob and Charlie then measure all the qubits they have in either the $\{|0\rangle, |1\rangle\}$ or $\{|+\rangle\, |-\rangle\}$ basis. Bob and Charlie use the outcome of these measurements to derive what values they know the key *cannot* be.

Message Signing: For a message b, Alice sends the signed message $(b, k_0^b k_1^b ... k_{L-1}^b)$.

Message Verification: To verify the signed message $(b, k_0^b k_1^b ... k_{L-1}^b)$, Bob (resp. Charlie) checks how many of the symbols in the signing key are in the set of values they eliminated from measurement and accepts if this number is smaller than $s_b L$ (resp. $s_c L$).

Because Alice does not know which qubits Bob measured and which he forwarded to Charlie (and vice versa), she gains no repudiation advantage from sending different states to the two receivers. Her best strategy is to send a signed message which has $L(s_c - s_b)L/2$ mismatches with the actual signature, which has a negligible chance of success in terms of L. There is also a negligible chance of forgery, as even if Bob faked the qubits he was forwarding on to Charlie, he would still need to guess the qubits that Charlie didn't forward on to him.

D. Removing the need for Authenticated Quantum Channels

2016 was when we finally saw quantum digital signatures proposed without authenticated quantum channels, in two papers: one by Yin, Fu and Chen [19] and one published a week later by Amiri et al. [20]. The latter paper will be the focus of this section.

This signature scheme relies on a key generation protocol; a protocol undertaken between Alice and each receiver independently. Based upon BB84 [5], the key generation protocol used between Alice and, for example, Bob is described below:

1. Bob generates qubits in either $|0\rangle, |1\rangle, |+\rangle$ or $|-\rangle$ states and forwards these states on to Alice.
2. Alice measures each state in either the X or Z basis, noting her chosen basis and results.
3. Alice and Bob publicly announce their choice of bases and throw out any results that were measured in different bases. Steps 1-3 are repeated until enough bits are produced.
4. Bob divides his key up into four parts of equal size:
   - The first set of bits, V, are all results from X basis measurements, and will be compared with the corresponding bits from Alice's measurements to see how much correlation there is between their bits.
   - The second set of bits, Z, are all results from Z basis measurements, and will be compared with Alice's corresponding bits to determine how much of their communications has been eavesdropped upon.
   - The third set of bits, C^B are bits that are forwarded on to Charlie.
   - The fourth set of bits, B are the bits that Bob keeps for his own private key. All other bits are discarded by both him and Alice.
5. If there is not enough correlation or the difference in the amount of correlation between the V and Z bits is too high, then the protocol is aborted.
6. Otherwise, Bob and Charlie exchange their parts of the keys through a secure & authenticated classical channel.

This protocol allows Alice to generate highly correlated keys with the recipients of her message. We use this to develop a digital signature scheme as follows, with $0 \leq s_b < s_c < 1/2$ and L as our security parameter:

Key Generation/Distribution: Bob and Charlie use the key generation protocol to generate two L-bit keys with Alice, half of each key being exchanged with the other party. Alice's copies of the keys are denoted A_0^B, A_1^B, A_0^C and A_1^C, where B (resp. C) denotes Bob (resp. Charlie). Bob and Charlie have their own keys with Alice B_0, B_1 & C_0, C_1 and their keys from each other B_0^C, B_1^C & C_0^B, C_1^B, respectively.

Message Signing: To sign a bit b, Alice sends the message (b, A_b^B, A_b^C).

Message Verification: To verify a sent message message (b, A_b^B, A_b^C), Bob checks the number of bits in A_b^B that mismatch B_b and the number of bits in A_b^C that mismatch B_b^C. If the total number of mismatches is below $s_b L/2$ then the message is rejected. Otherwise the message is accepted.

This signature is a more general version of a protocol from 2015. While the protocol described above uses a key generation protocol that creates correlated bit strings between parties, protocol P2 in [17] uses already shared keys which perfectly match.

Like the protocol in SEC. V C, security against repudiation is offered as Alice does not know which bits of their respective keys Bob and Charlie passed on to each other.

For Bob to forge a message to send to Charlie, he needs to guess the values of the bits that Charlie did not forward on to him, which is negligible for the same reason as in SEC. V C. Bob has the highest chance of success if he eavesdrops on the key generation protocol between Alice and Charlie, which he now can given that the channel is not authenticated. If Bob regularly guesses the wrong qubit to forward to Alice, this will result in low correlations between Alice and Charlie's measurements in the X or Z basis. Thus when Alice and Charlie look at the correlations between their V and X parts of their keys, they will either find low correlations in V or a significant difference between the number of correlations in V and the number of correlations in X. Either way, the protocol will be aborted. Thus the probability of a successful forgery is negligible.

VI. EXPERIMENTAL ACHIEVEMENTS

VII. CONCLUSION AND DISCUSSION

[1] R. L. Rivest, A. Shamir, and L. Adleman, Commun. ACM **21**, 120 (1978).

[2] P. W. Shor, SIAM Review **41**, 303 (1999).

[3] E. Martin-Lopez, A. Laing, T. Lawson, R. Alvarez, X.-Q. Zhou, and J. L. O'Brien, Nat. Photon. **6**, 773 (1999).

[4] R. Dridi and H. Alghassi, "Prime factorization using quantum annealing and computational algebraic geometry," (2016), arXiv:1604.05796.

[5] C. H. Bennett and G. Brassard, in *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing* (1984) pp. 175–179.

[6] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, Journal of Cryptology **5**, 3 (1992).

[7] `http://www.idquantique.com/`.

[8] D. Gottesman and I. Chuang, "Quantum digital signatures," (2001), arXiv:quant-ph/0105032.

[9] L. Lamport, Technical Report SRI-CSL-98 (1979).

[10] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, Phys. Rev. Lett. **87**, 167902 (2001).

[11] A. S. Holevo, Problems of Information Transmission **9**, 3 (1973).

[12] W. K. Wootters and W. H. Zurek, Nature **299**, 802 (1982).

[13] H. Barnum, C. Crepeau, D. Gottesman, A. Smith, and A. Tapp, in *Foundations of Computer Science, 2002. Proceedings. The 43rd* (2002) pp. 449–458.

[14] V. Dunjko, P. Wallden, and E. Andersson, Phys. Rev. Lett. **112**, 040502 (2014).

[15] I. Ivanovic, Physics Letters A **123**, 257 (1987).

[16] Note that if all parties are honest, the state at both null outputs is the vacuum state, so few if any photons should be seen.

[17] P. Wallden, V. Dunjko, A. Kent, and E. Andersson, Phys. Rev. A **91**, 042304 (2015).

[18] S. Bandyopadhyay, R. Jain, J. Oppenheim, and C. Perry, Phys. Rev. A **89**, 022336 (2014).

[19] H.-L. Yin, Y. Fu, and Z.-B. Chen, Phys. Rev. A **93**, 032316 (2016).

[20] R. Amiri, P. Wallden, A. Kent, and E. Andersson, Phys. Rev. A **93**, 032325 (2016).

Appendix A: Arbitrated Quantum Signatures