Multiparty quantum signature schemes

Quantum Information and Computation, Vol. Vol., No. No. (Year) Page Nos.

MULTIPARTY QUANTUM SIGNATURE SCHEMES

JUAN MIGUEL ARRAZOLA
Institute for Quantum Computing and Department of Physics, University of Waterloo, 200 University Avenue West, Waterloo, ON, N2L 3G1, Canada

PETROS WALLDEN
School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, UK

ERIKA ANDERSSON
SUPA, Institute of Photonics and Quantum Sciences, Heriot-Watt University, Edinburgh, EH14 4AS, UK

Received June 9, 2015
Revised January 19, 2016

Digital signatures are widely used in electronic communications to secure important tasks such as financial transactions, software updates, and legal contracts. The signature schemes that are in use today are based on public-key cryptography and derive their security from computational assumptions. However, it is possible to construct unconditionally secure signature protocols. In particular, using quantum communication, it is possible to construct signature schemes with security based on fundamental principles of quantum mechanics. Several quantum signature protocols have been proposed, but none of them has been explicitly generalised to more than three participants, and their security goals have not been formally defined. Here, we first extend the security definitions of Swanson and Stinson [1] so that they can apply also to the quantum case, and introduce a formal definition of transferability based on different verification levels. We then prove several properties that multiparty signature protocols with information-theoretic security – quantum or classical – must satisfy in order to achieve their security goals. We also express two existing quantum signature protocols with three parties in the security framework we have introduced. Finally, we generalize a quantum signature protocol given in [2] to the multiparty case, proving its security against forging, repudiation and non-transferability. Notably, this protocol can be implemented using any point-to-point quantum key distribution network and therefore is ready to be experimentally demonstrated.

Keywords: Quantum Cryptography, Quantum Communication, Quantum Key Distribution
Communicated by: S Braunstein & H Zbinden

1 Introduction

Digital signatures are important cryptographic building blocks which are widely used to provide security in electronic communications. They achieve three main cryptographic goals: authentication, non-repudiation, and transferability. These properties make them suitable for

securing important tasks such as financial transactions, software updates, and legal contracts. The digital signatures schemes that are in use today, which are based on public-key cryptography, derive their security from unproven computational assumptions, and most of them – notably those based on the RSA algorithm or on elliptic curves – can be broken by quantum computers [3].

Consequently, from both a practical and fundamental perspective, there has been an interest in studying signature protocols that do not rely on computational assumptions, but instead offer information-theoretic security. These schemes were first introduced by Chaum and Roijakkers [4] and are known as *unconditionally secure signature* (USS) schemes. Besides the proposal of Chaum and Roijakkers, several other USS protocols have been suggested [5, 6, 7, 8, 9, 10, 11, 12, 2], most of them based on removing standard trust assumptions from message authentication codes (MACs). However, most known classical USS protocols proposed so far rely on the assumption of either a trusted arbiter or authenticated broadcast channels, and crucially, all of them require the use of secure channels, which are impossible to realize, practically, with information-theoretic security using only classical communication [13, 14].

Once quantum communication is allowed, it becomes possible to construct signature schemes whose information-theoretic security is based on fundamental principles of quantum mechanics. These are known as quantum signature (QS) schemes. The first QS protocol was proposed by Gottesman and Chuang [15], who introduced the main ideas for bringing digital signatures into the quantum world. Although influential from a fundamental point of view, their scheme requires the preparation of complex quantum states, performing quantum computation on these states and storing them in quantum memory, making the protocol highly impractical. This is also an issue for other protocols that appeared shortly after [16, 17].

Recently, new QS protocols that do not require a quantum memory and which can be realized with standard quantum-optical techniques have been proposed [2, 18, 19]. Some of these protocols have also been demonstrated experimentally [20, 21], thus establishing their viability as a practical technology. A short review of these developments can be found in Ref. [22]. Nevertheless, these schemes have not been generalized to more than three participants, and their security goals have not been formally defined. Overall, a security framework for quantum signature schemes that includes rigorous definitions of security suitable for multiparty protocols has not yet been proposed. In the absence of such a framework, it is not clear how to design secure multiparty protocols nor what the concrete advantages of quantum signatures are compared to their classical counterparts.

In this work, we provide a security framework suitable for USS protocols involving an arbitrary number of participants. We follow the definitions by Swanson and Stinson [1], generalizing them so that they can apply also to the quantum case, and introduce a formal definition of transferability based on different verification levels. This notion of transferability applies to both quantum and classical protocols. We also present a characterization of the general structure of USS protocols and introduce rigorous definitions of security. Additionally, we prove several properties that these protocols must satisfy in order to achieve their security goals. We then express two existing protocols for quantum signatures with three parties within the framework we developed. Finally, we make use of our results to generalize a quantum protocol of Wallden et. al [2] to the multiparty case and prove its security against forging,

repudiation and non-transferability. Notably, this protocol can be implemented using any point-to-point quantum key distribution network and therefore is ready to be experimentally demonstrated.

2 Definitions for USS Protocols

A QS protocol is carried out by a set of participants and is divided into two stages: the *distribution* stage and the *messaging* stage. The distribution stage is a quantum communication stage, where the parties exchange quantum and classical signals according to the rules of the protocol. Although in principle they could store the received quantum states in a quantum memory, we focus on more practical protocols in which the participants perform measurements on the states and store the outcomes in a classical memory. The participants may also process their data and communicate classically with each other. Overall, each participant is left with a set of rules for signing messages and for verifying signatures. These rules generally depend on their measurement outcomes and the classical communication. At the end of the distribution stage, the parties decide whether to continue to the messaging stage or to abort the protocol. In the messaging stage, one of the participants (the signer) signs a message by attaching a classical string (the signature) to the message. When a participant receives a signed message, they verify its validity according to the rules of the protocol.

A USS protocol must achieve authenticity, non-repudiation, and transferability as its main security goals. Informally, these goals can be defined as follows:

1. Authentication: Except with negligible probability, an adversary cannot create a message and signature pair that is accepted by another participant, i.e. a signature cannot be forged.

2. Non-repudiation: Except with negligible probability, a signer cannot later successfully deny having signed a message that has been accepted by an honest recipient.

3. Transferability: A recipient that accepts a signed message can be confident that, except with negligible probability, the signature will also be accepted by other participants.

In order to satisfy non-repudiation and transferability, each recipient must have a method of determining whether other participants will also agree on the validity of a signature. This is straightforward in classical public-key schemes, since every recipient applies the same rule to verify a signature. However, as we discuss later in this paper, in an information-theoretic scenario, every recipient must have a different rule to verify a signed message – or, at least, two participants must have the same verification algorithm with low probability*. Thus, a security model for USS schemes must deal carefully with the notion of non-repudiation and the transferability of signatures.

*Following Swanson and Stinson [1], with "verification algorithm", we understand a full specification of the rules an individual participant is using to verify a message. For example, different recipients could use a more generic "verification function", which is the same for all participants, but with random inputs which differ for different recipients. What we mean by the verification algorithm of an individual participant would, in this case, be the generic verification function together with that participant's specific random inputs. This way of defining the recipients' verification functions makes sense considering that a recipient might in this example know neither what what the underlying generic function is, nor what the random inputs are, only what the resulting combination of the generic verification function and random inputs is. This definition of verification function also makes sense for quantum signature protocols.

We now generalize the work of Swanson and Stinson [1] in the context of USS schemes to construct formal definitions that are also suitable for quantum signature schemes and allow for different levels of verification. This will permit us to formalize the structure of general USS protocols, provide rigorous security definitions, and illustrate properties they must possess in order to be secure.

Definition 1 *A USS protocol $\mathcal{Q}$ is an ordered set $\{\mathcal{P}, X, \Sigma, L, \mathrm{Gen}, \mathrm{Sign}, \mathrm{Ver}\}$ where:*

- *The set $\mathcal{P} = \{P_0, P_1, \ldots, P_{N-1}\}$, is the set of N different participants involved in the protocol. We fix P_0 to be the signer, and P_i are the possible recipients, with $i \in \{1, \cdots, N-1\}$. X is the set of possible messages and Σ is the set of possible signatures.*

- Gen *is the generation algorithm that gives rise to the functions* Sign *and* Ver *that are used to generate a signature and verify its validity. More precisely, the generation algorithm specifies the instructions for the quantum and classical communication that takes place in the distribution stage of the protocol. Additionally, the generation algorithm instructs how to construct the functions* Sign *and* Ver *based on the data obtained during the distribution stage. The generation algorithm includes the option of outputting an instruction to abort the protocol.*

- *The signature function* Sign *is a deterministic function $X \to \Sigma$ that takes a message x and outputs a signature $\sigma \in \Sigma$.*

- *$L = \{-1, 0, 1, \ldots, l_{\max}\}$ is the set of possible verification levels of a signed message. A verification level l corresponds to the minimum number of times that a signed message can be transferred sequentially to other recipients. For a given protocol, the maximum number of sequential transfers that can be guaranteed is denoted by $l_{\max} \leq N-1$.*

- *The verification function* Ver *is a deterministic function $X \times \Sigma \times \mathcal{P} \times L \to \{\mathrm{True}, \mathrm{False}\}$ that takes a message x, a signature σ, a participant P_i and a level l, and gives a truth value depending on whether participant P_i accepts the signature as valid at the verification level l. We denote a verification function with a fixed participant P_i and level l as $\mathrm{Ver}_{i,l}(x, \sigma) := \mathrm{Ver}(x, \sigma, i, l)$.*

In general, the generation algorithm must involve randomness in the construction of the signing and verification functions. The randomness may be generated locally by each participant or it can be generated and distributed by a trusted third party. It can arise from the intrinsic randomness of quantum measurements, or by other means. Therefore, even though the signing and verification functions are deterministic functions, they are randomly generated. An illustration of the distribution stage for a generic USS protocol can be seen in Fig. 1.

The verification levels are a method of determining whether a signature can be transferred sequentially among participants. As an illustration, consider a protocol involving a signer Alice, a recipient Bob, and a bank. Other participants may be involved as well. Bob receives a payment from Alice which is signed using a USS protocol, and Bob wants to transfer this signed message to the bank. For Bob, it does not suffice to verify that the signature comes

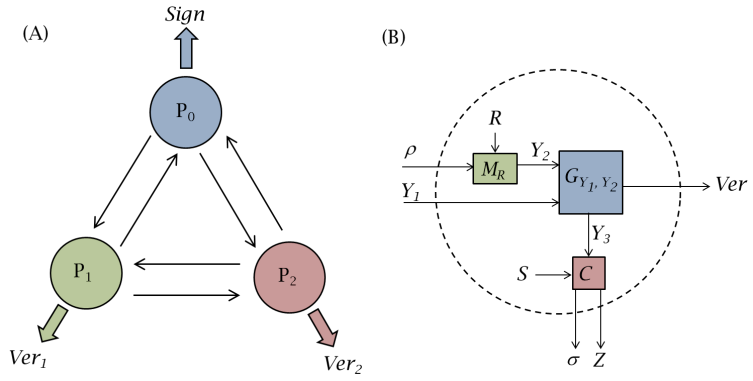

Fig. 1. (A) Schematic portrayal of a possible generation algorithm in the distribution stage of a QS protocol with three participants. The three parties exchange messages over classical and quantum channels. At the end of their communication, the signer has a specification of the signing algorithm, and the recipients have a specification of their respective verification functions. (B) An example of a generation algorithm for one of the recipients. From their perspective, they receive a quantum state ρ and a classical message Y_1 from the other participants. A measurement M_R that depends on a random variable R is carried out on the quantum state, and the outcome Y_2, together with the classical data Y_1, is fed to an algorithm G_{Y_1,Y_2}. This program outputs data Y_3 that, together with another possibly random variable S, is fed to a second algorithm C that determines the quantum and classical messages sent to the other participants. After several iterations of these steps, the program G_{Y_1,Y_2} outputs the verification function.

from Alice and that she cannot repudiate it – he also needs a guarantee that when he transfers the signed message to the bank, they will be able to validate it. Now suppose that the bank also requires the ability to transfer the message to another participant, otherwise they don't accept the message. Then Bob needs a guarantee that it can be transferred *twice* in sequence, from himself to the bank and from the bank to another participant. In general, Bob may require that a signed message be transferred many times in sequence. This guarantee is provided by the verification levels: With high probability, a signature that is verified at level l can be transferred l times in sequence. A signature that is verified at level $l = 0$ is certified to have come from the signer, but does not have a guarantee that it can be transferred to other participants. The role of the verification level $l = -1$ is to prevent repudiation, as will be explained in section 2.

We now introduce additional useful definitions, which are inspired by Ref. [1] and generalized to allow different levels of verification. As a starting point, it is important that a USS protocol works properly when all parties are honest.

Definition 2 *A USS protocol $\mathcal{Q}$ is* correct *if* $\mathrm{Ver}_{(i,l)}(x, \mathrm{Sign}(x)) = \mathrm{True}$ *for all* x, i, l.

Since USS protocols have different verification functions for different participants as well as different levels of verification, it is important to carefully specify what it means for a particular signature to be valid.

Definition 3 *A signature σ on a message x is* authentic *if* $\sigma = \mathrm{Sign}(x)$.

Definition 4 *A signature σ on a message x is* valid *if* $\mathrm{Ver}_{(i,0)}(x,\sigma) = \mathrm{True}$ *for all* $i \in \{1,\cdots,N-1\}$.

Thus, a valid signature is simply one for which all participants can verify that it originates from the intended signer. Crucially, a valid signature does not need to be authentic, a possibility not originally considered in Ref. [1].

Definition 5 *A signature σ on a message x is* i-acceptable *if* $\mathrm{Ver}_{(i,0)}(x,\sigma) = \mathrm{True}$.

Note that, as opposed to a valid signature, an i-acceptable signature may not pass the verification functions of participants other than P_i. Therefore, an i-acceptable signature may not be a valid signature.

Definition 6 *A signature σ on a message x is* i-fraudulent*, if σ is i-acceptable but not valid.*

As discussed before, the participants may additionally be interested in the transferability of the signature. This motivates the following definitions.

Definition 7 *A signature σ on a message x is* l-transferable *if* $\mathrm{Ver}_{(i,l)}(x,\sigma) = \mathrm{True}$ *for all* $i \in \{1,\cdots,N-1\}$ *and there exists j such that* $\mathrm{Ver}_{(j,l+1)}(x,\sigma) = \mathrm{False}$*. For $l = l_{\max}$, the function* $\mathrm{Ver}_{(j,l_{\max}+1)}(x,\sigma)$ *is not defined and we assume by convention that it is always* False*.*

The above definition means that a signature is l-transferable if l is the largest level for which this signature will pass the verification test of all participants.

Definition 8 *A signature σ on a message x is* (i,l)-transferable *if* $\mathrm{Ver}_{(i,l)}(x,\sigma) = \mathrm{True}$ *and* $\mathrm{Ver}_{(i,l+1)}(x,\sigma) = \mathrm{False}$.

Thus, an (i,l)-transferable signature will pass the verification test of participant i at level l, but not at any other higher level. As opposed to an l-transferable signature, it may not pass the verification functions of other participants.

2.1 Dispute Resolution

In traditional digital signature schemes based on public-key cryptography, there is a public verification function to test the validity of a signature. If a person denies having signed a message, the recipient who initially verified the signature can show it to other honest parties – a judge for example – who will use the same public verification function to certify its validity and therefore reject the signer's claims.

However, as we show in section 3, in a USS scheme different participants have different verification functions, which makes it possible in principle for two or more participants to disagree on the validity of a signature. The mechanism to prevent repudiation must take this into account. Suppose that Alice signs a contract and sends it to Bob, who uses his verification function to verify the signature. The signature passes his verification test at level $l = 0$ and he is convinced that the message comes from Alice. Later, Alice attempts to repudiate by denying that she signed the contract. Bob knows that the other participants have different verification functions than his own, so what can be done to prevent Alice from repudiating? Non-repudiation is ensured by incorporating a dispute resolution method: a mechanism to handle the event of a disagreement on the validity of a signature. It is expected that dispute resolution will be invoked relatively rarely. It is akin to an appeal procedure which is very expensive for the participant who loses. Any participant (honest or dishonest), who thinks he might lose the dispute resolution, will avoid any action that could lead to someone invoking

it. Therefore, while the dispute resolution may seem complicated and resource-expensive in terms of communication, is not something that affects the effectiveness of the protocol, since any rational participant, whether adversarial or honest, will always take actions that guarantee that any other rational participant would not invoke dispute resolution. Based on Ref. [1], we formally define such a dispute resolution method as follows.

Definition 9 *A dispute resolution method* DR *for a USS scheme $\mathcal{Q}$ is a procedure invoked whenever there is a disagreement on whether a signature σ on a message x is a valid signature originating from the signer P_0. The participant invoking the dispute resolution can be anyone, including the signer P_0. The procedure consists of an algorithm* DR *that takes as input a message-signature pair (x, σ) and outputs a value* {Valid, Invalid} *together with the rules:*

1. *If* $\mathrm{DR}(x, \sigma)$ *outputs* Valid*, then all users must accept (x, σ) as a valid signature for x.*
2. *If* $\mathrm{DR}(x, \sigma)$ *outputs* Invalid*, then all users must reject (x, σ) as a valid signature for x.*

Defining a particular dispute resolution method constitutes a crucial part of specifying a USS protocol. Whether a protocol is secure against repudiation will generally depend on the choice of dispute resolution. But what are the concrete possibilities that we can choose from? A simple strategy is to designate a trusted participant to be in charge of deciding the validity of a signature whenever the dispute resolution method is invoked. This participant, who may have access to more information about the protocol than others, serves as an arbiter who has the final word whenever there is a dispute. An obvious drawback of this choice is the necessity of trust: If the arbiter behaves dishonestly, perhaps due to being blackmailed to do so, the entire security of the protocol is compromised. In this paper, we focus on a *majority vote* dispute resolution method.

Definition 10 *When the validity of a message-signature pair (x, σ) is invoked, a majority vote dispute resolution method* $\mathrm{MV}(x, \sigma)$ *is defined by the following rule:*

1. $\mathrm{MV}(x, \sigma) = \mathrm{Valid}$ *if* $\mathrm{Ver}_{(i,-1)}(x, \sigma) = \mathrm{True}$ *for more than half of the users.*
2. $\mathrm{MV}(x, \sigma) = \mathrm{Invalid}$ *otherwise,*

where $\mathrm{Ver}_{(i,-1)}$ *is the verification function at level $l = -1$.*

The need for a verification level $l = -1$ can be understood as a mechanism to prevent repudiation by Alice, and it is only relevant when DR is invoked. Intuitively, $\mathrm{Ver}_{(i,-1)}$ should be chosen such that is infeasible to produce a signature that passes the verification function of one participant at level $l = 0$, but does not pass the verification function of the majority of participants at level $l = -1$. This will be formalized in section 3.

The majority vote dispute resolution method was implicitly used in the protocols of [2, 18] when discussing security against repudiation. The obvious advantage of the majority vote method is that we do not need to trust any fixed participant, but instead assume only that at least *most* of them are not dishonest. However, we emphasize that the security definitions of the following section do not depend on a particular choice of DR.

Note that a dispute resolution method can be used by any participant to convince others of the validity of a signature, even when the signature is only verified at level $l = 0$. If the

protocol is secure against repudiation – as will be formally defined in the next section – then no person other than the signer will be able to create a signature that is deemed valid by the dispute resolution method. Therefore, if DR is invoked and outputs "Valid", everyone is already convinced that the signature must have come from the signer. This means that the verification levels serve the specific purpose of providing the participants with an assurance that other people will sequentially verify a transferred signature *without the need to invoke dispute resolution.* This is desirable because carrying out dispute resolution may be expensive and should only be invoked under special circumstances.

Finally, we also consider the case in which a participant is dishonest about the level at which they verify a signature. For instance, suppose that Bob wants to transfer a message regarding a payment by Alice, signed by Alice, to a store. The store only accepts signatures that they can transfer to a bank, so Bob needs an assurance that Alice's signature can be transferred twice in sequence. Bob verifies the signature at level $l = 2$ and sends it to the store. The store, however, is dishonest, and lies to Bob by claiming that they verified the signature only at level $l = 0$, even though Bob knows that they should have verified it at least at level $l = 1$. If the protocol is secure against repudiation, Bob can invoke dispute resolution to make everyone, including the bank, agree on the validity of the signature. But in order to resolve disputes regarding the verification level of a signature, we need an additional dispute resolution method.

Definition 11 *A transferability dispute resolution method at level l,* TDR*, for a QS scheme $\mathcal{Q}$, consists of an algorithm* $\mathrm{DR_l}$ *that takes as input a message-signature pair (x, σ) and verification level l and outputs* $\{l-$transferable, not $l-$transferable$\}$ *together with the rules:*

1. *If* $\mathrm{DR_l}(x, \sigma, l)$ *outputs* $l-$transferable*, then all users must accept (x, σ) as an l-transferable signature for x.*
2. *If* $\mathrm{DR_l}(x, \sigma, l)$ *outputs* not l-transferable*, then all users must reject (x, σ) as an l-transferable signature for x.*

For this form of dispute resolution method, we can also use a majority vote method defined as before.

Definition 12 *A majority vote transferability dispute resolution method at level l,* $\mathrm{MV}(x, \sigma, l)$*, is defined by the following rule:*

1. $\mathrm{MV}(x, \sigma, l) = l-$transferable *if* $\mathrm{Ver}_{(i,l-1)}(x, \sigma) =$ True *for more than half of the users.*
2. $\mathrm{MV}(x, \sigma, l) =$ not $l-$transferable *otherwise.*

If the protocol offers transferability, as will be formally defined in the next section, any participant who verifies a signature at level l has a guarantee that, with high probability, any other participant will verify the signature at level at least $l - 1$. Therefore, if the majority of participants are honest, a majority vote will indeed deem the signature that was verified at level l by an honest participant as an $(l - 1)$-transferable signature. This form of dispute resolution can serve as a deterrent for dishonest behaviour. In our previous example, the store is discouraged from lying to Bob as they know that a transferability dispute resolution can be used to detect their dishonesty, for which they can be penalized.

2.2 Security definitions

Previously, we introduced the security goals of USS schemes. We are now in a position to define them formally. More than one of the participants can be malevolent, so in general we must look at coalitions of participants that attack the scheme. In an attempt at repudiation, the coalition must include the signer, whereas a coalition aiming to forge a signature does not include the signer. Formally, we define successful cases of repudiation and forging as follows:

Definition 13 *Given a USS protocol $\mathcal{Q}$ and a coalition $C \subset \mathcal{P}$ of malevolent participants – including the signer P_0 – that output a message-signature pair (x, σ), we define* repudiation *to be the function:*

$$Rep_C(\mathcal{Q}, DR, \sigma, x) = \begin{cases} 1 & \text{if } (\sigma, x) \text{ is } i\text{-acceptable for some } i \notin C \text{ and } \mathrm{DR}(\sigma, x) = \mathrm{Invalid} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Thus, a coalition succeeds at repudiation if they can produce a signature that passes the verification test of one of the honest participants at level $l = 0$, but when a DR is invoked, it will be decided that the signature is invalid. According to this definition, a malevolent signer may be able to repudiate with respect to some dispute resolution method, but not other methods.

Definition 14 *Given a USS protocol $\mathcal{Q}$ and a coalition of malevolent parties $C \subset \mathcal{P}$ – not including the signer P_0 – that output a message-signature pair (x, σ), we define* forging *to be the function:*

$$Forg_C(\mathcal{Q}, \sigma, x) = \begin{cases} 1 & \text{if } (\sigma, x) \text{ is } i\text{-acceptable for some } i \notin C \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

A successful forgery therefore only requires the coalition to create a signature that passes the verification test of *one* honest participant at level $l = 0$. Note that we could have additionally asked that the signature be deemed valid by the DR method, but that would constitute a more difficult task for the attackers.

Definition 15 *Given a USS protocol $\mathcal{Q}$, a coalition of malevolent parties $C \subset \mathcal{P}$ – including the signer P_0 – that output a message-signature pair (x, σ), and a verification level l, we define* non-transferability *to be the function:*

$$NonTrans_C(\mathcal{Q}, \sigma, x, l) = \begin{cases} 1 & \text{if } \mathrm{Ver}_{(i,l)}(\sigma, x) = \mathrm{True} \text{ for some } i \notin C \text{ and } \mathrm{Ver}_{(j,l')}(\sigma, x) = \mathrm{False} \\ & \text{for some } 0 \leq l' < l \text{ and some } j \neq i,\ j \notin C \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Therefore, a message-signature pair will be non-transferable at level l if the coalition can produce a signature that at least one honest participant verifies at level l, but some other honest participant does not verify at a lower level. Thus, if the signature is non-transferable,

there exists a sequence of participants such that, as the signature is transferred in the order of the sequence, at least one of them will not agree that he can transfer the signature to the remaining participants.

We can now state the main security definitions for USS protocols:

Definition 16 *Given a coalition $C \subset \mathcal{P}$, a USS protocol $\mathcal{Q}$ is called* ϵ-secure against forging *if, using their optimal strategy, the probability that the coalition outputs a message-signature pair (x, σ) constituting a successful forgery satisfies*

$$\Pr[Forg_C(\mathcal{Q}, \sigma, x) = 1] \leq \epsilon, \tag{4}$$

where the probability is taken over any randomness in the generation algorithm and the optimal forging strategy.

Definition 17 *Given a coalition $C \subset \mathcal{P}$ and a dispute resolution method* DR*, a USS protocol $\mathcal{Q}$ is called* ϵ-secure against repudiation *if, using their optimal strategy, the probability that the coalition outputs a message-signature pair (x, σ) constituting successful repudiation satisfies*

$$\Pr[Rep_C(\mathcal{Q}, \sigma, x) = 1] \leq \epsilon, \tag{5}$$

where the probability is taken over any randomness in the generation algorithm and the optimal repudiation strategy.

Definition 18 *Given a coalition $C \subset \mathcal{P}$, a USS protocol $\mathcal{Q}$ is called* ϵ-transferable *at level l if, using their optimal strategy, the probability that the coalition outputs a non-transferable message-signature pair (x, σ) at level l satisfies*

$$\Pr[NonTrans_C(\mathcal{Q}, \sigma, x, l) = 1] \leq \epsilon, \tag{6}$$

where the probability is taken over any randomness in the generation algorithm and the optimal cheating strategy.

Note that the notion of transferability only makes sense between honest participants. As discussed before, even if the protocol is ϵ-transferable, if a participant transfers a signed message to a dishonest participant, the dishonest person can always deny that they have an assurance of being able to transfer it further. In that case, a transferability dispute resolution method can be invoked at level l.

Finally, we note that the security definitions we have provided here can in principle be adapted or relaxed, depending on the particular scope of the protocol. For example, depending on the context, it may or may not be useful to be able to cheat with just any recipient, without knowing specifically who this is. For example, a forger may want a message to be accepted specifically by a bank, and it may be of no interest that the message is accepted by another unknown user out of many possible ones. Thus, schemes offering other types of security, such as sufficiently low probability for forging a message with a particular recipient, should not be completely ruled out.

3 Properties of USS protocols.

In this section, we examine several required properties of USS protocols. Understanding these properties is important for several reasons. First, they serve as guiding principles for the construction of new protocols. Additionally, from a fundamental point of view, they provide insight regarding precisely what characteristics of USS protocols give rise to their security. Finally, delineating these properties allows us to construct a coherent picture of the practical challenges to building these protocols as well as their advantages and limitations compared to signature schemes with computational security. In the remainder of this section, we list several of these properties and, whenever relevant, prove that they are required for the security of USS protocols.

Observation 1 *In any secure USS protocol, all classical communication must be authenticated.*

First, authentication is necessary as a guarantee that the participants of the protocol are who they are supposed to be. Otherwise, it would be possible for unauthorized outsiders to participate and compromise the security of the protocol, for example during dispute resolution. Moreover, just as with quantum key distribution, without authentication any USS protocol is subject to a man-in-the-middle-attack, where an attacker impersonates one or more participants, thus rendering the entire scheme insecure. Information-theoretic authentication requires shared secret keys, so the above observation implies that any secure USS protocol requires secret keys shared between the participants, of length proportional to the logarithm of the length of the messages sent [23].

Observation 2 $\text{Ver}_{(i,l)}(x,\sigma) = \text{True} \Rightarrow Ver_{(i,l')}(x,\sigma) = \text{True}$ *for all* $l' < l$.

Since the verification level of a signature corresponds to the maximum number of times a signature can be transferred, a signature that is verified at a given level should also be verified at all lower levels.

We have mentioned before that in an information-theoretic scenario, it is necessary that each participant has a different verification function with high enough probability. We now show this explicitly, following Ref. [1].

Observation 3 *[1] For any USS protocol that is* ϵ*-secure against forging, it most hold that*

$$\Pr\left(\text{Ver}_{(i,l)} \neq \text{Ver}_{(j,l)}\right) \geq 1 - \epsilon \tag{7}$$

for all l *and for all* $i \neq j$.

Proof. If $\text{Ver}_{(i,l)} = \text{Ver}_{(j,l)}$, then participant P_i can conduct an exhaustive search for a message-signature pair such that $\text{Ver}_{(i,l)}(x,\sigma) = \text{True}$. But since $\text{Ver}_{(i,l)} = \text{Ver}_{(j,l)}$, participant P_i will also have produced a message-signature pair that passes the verification function of participant P_j. From observation 2, if participant P_i can produce such a signature, he can also produce a signature such that $\text{Ver}_{(j,0)}(x,\sigma) = \text{True}$, which constitutes successful forging. Therefore, the verification functions must be different at all levels to guarantee security against forging. If the protocol is ϵ-secure against forging, then this should only happen with probability smaller than ϵ. □

Here we should remark that it is possible for probabilistic protocols, in particular quantum signature protocols, to have two participants with the same verification functions, but the probability of this happening must be made small enough for the protocol to be secure. Alternatively, one could consider other security models in which this condition is relaxed. For example, that two participants may have the same verification function with higher probability, but it is unlikely for a cheating party to know who might have the same function. More generally, as further discussed below, there will be conditions not only on the probability that two participants have the same verification function, but also that it should be hard for a participant to guess the verification function of another participant.

Corollary 1 *A secure USS protocol with a finite number of possible signatures can only exist for a finite number of participants.*

Proof. For a given verification level l and message x, a verification function for participant P_i is equivalent to the specification of a subset $S \subset \Sigma$ of signatures such that $\mathrm{Ver}_{(i,l)}(x, \sigma) =$ True. Since the possible number of signatures is a finite set, so is the number of verification functions. From Observation 3, in any secure protocol, every participant must have a different verification function with high probability, and since there is only a finite number of these functions, there can only be a finite number of participants. □

In principle, one could add new participants to the protocol by using further communication between the new participant and the original ones. Essentially, in order to construct a protocol with $N + 1$ participants from a protocol with N participants, the new participant could interact with all others in exactly the same way as if he had participated directly in a protocol with $N + 1$ participants. This interaction could happen at a later time than the original distribution stage.

Observation 4 *The generation algorithm of a secure USS protocol must randomly generate the verification and signing functions.*

Proof. If all functions are generated deterministically, then the specification of the protocol is sufficient for every participant to know the signing function and all the verification functions. However, if a participant knows the signing algorithm, forging is trivial since he can produce authentic signatures. Similarly, if a participant knows the verification function of another person, he can conduct an exhaustive search to find a message-signature pair that is validated by the other participant, which constitutes a successful forgery. Finally, if a signer knows the verification function of the other participants, she can conduct an exhaustive search to find a signature that is accepted by one of them at level l, but rejected by everyone else at level $l - 1$, which allows her to repudiate or break transferability. Thus, a secure protocol requires a randomized generation algorithm.□

The randomness in the protocol may be produced locally by each participant, or it may be generated and distributed by a trusted third party. The randomness may arise from the intrinsic randomness of performing measurements on quantum systems, or by other means. Overall, from the point of view of each participant, the generation algorithm must induce a probability distribution over the possible signing functions as well as the possible verification functions. Therefore, the security of a USS protocol depends crucially on the difficulty of

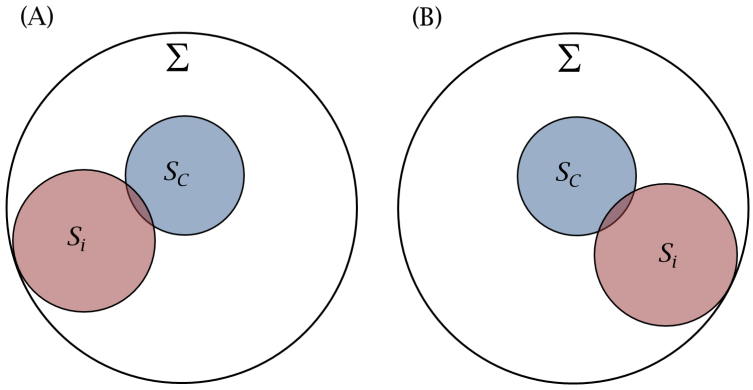

Fig. 2. S_C is the set of signatures that pass the verification functions at level $l = 0$ of all members of a coalition C. S_i is the set of signatures that pass the verification function at level $l = 0$ of a participant P_i outside of a coalition. If the protocol is secure against repudiation, the intersection $S_C \cap S_i$ must be small compared to S_C. More generally, the coalition cannot determine what the set S_i is, except with low enough probability. For example, if the protocol is secure against forging, the coalition should not be able to distinguish whether they are in situation (A) or (B).

guessing the functions of other participants. We can formalize this requirement with the following observations.

Observation 5 *For a given message x, let S_C be the set of signatures that pass the verification functions at level $l = 0$ of all members of a coalition C. Similarly, let S_i be the set of signatures that pass the verification function at level $l = 0$ of a participant P_i outside of the coalition. Then, for any USS protocol that is ϵ-secure against forging, it must hold that*

$$\frac{|S_i \cap S_C|}{|S_C|} \leq \epsilon \text{ for all } i \notin C, \tag{8}$$

where $|S|$ is the size of a set S and $S_i \cap S_C$ is intersection between S_i and S_C.

Proof. Let (x, σ_c) be a message-signature pair drawn uniformly at random from S_C. If this signature passes the verification function at level $l = 0$ of a participant outside of the coalition, it will constitute a successful forgery. The probability that this happens is given by $\frac{|S_i \cap S_C|}{|S_C|}$, which must be smaller than ϵ in order for the protocol to be ϵ-secure against forging. □

An illustration of the above property can be seen in Figure 2. Notice that if a protocol is correct, authentic signatures are verified by all participants. Therefore, for correct protocols it holds that $S_C \cap S_i \neq \emptyset$. Similarly to the above, we can provide a condition for security against repudiation.

Observation 6 *For a given message x, let S_i be the set of signatures that pass the verification function at level $l = 0$ of a participant P_i outside of a coalition C, and let Σ be the set of all possible signatures for this message. Then, for any USS protocol that is ϵ-secure against*

forging and ϵ'-secure against repudiation with a majority vote dispute resolution, it must hold that

$$\frac{|S_i|}{|\Sigma|} \leq \frac{\epsilon'}{1-\epsilon}. \tag{9}$$

Proof. Let σ_r be a signature drawn uniformly at random from the set Σ of possible signatures. The probability that the signer can repudiate with this signature is given by

$$\begin{aligned}\Pr(Rep) &= \Pr[\mathrm{Ver}_{(i,0)}(x,\sigma_r) = \text{True AND } \mathrm{MV}(x,\sigma_r) = \text{Invalid}] \\ &= \Pr[\mathrm{MV}(x,\sigma_r) = \text{Invalid}|\mathrm{Ver}_{(i,0)}(x,\sigma_r) = \text{True}] \times \Pr[\mathrm{Ver}_{(i,0)}(x,\sigma_r) = \text{True}]. \end{aligned} \tag{10}$$

If σ_r is drawn uniformly at random from Σ, conditioning on σ_r passing the verification function of participant P_i induces a uniform distribution over the set S_i. From observation 5, if the protocol is ϵ-secure against forging, the probability that a signature drawn uniformly at random from S_i passes the verification function of another honest participant must be smaller than or equal to ϵ. Consequently, the probability that a signature drawn randomly from S_i passes the verification function of the *majority* of participants must also be smaller than ϵ, so we have that

$$\Pr[\mathrm{MV}(x,\sigma_r) = \text{Valid}|\mathrm{Ver}_{(i,0)}(x,\sigma_r) = \text{True}] \leq \epsilon$$

and therefore

$$\begin{aligned}\Pr[\mathrm{MV}(x,\sigma_r) = \text{Invalid}|\mathrm{Ver}_{(i,0)}(x,\sigma_r) = \text{True}] &= 1 - Pr[\mathrm{MV}(x,\sigma_r) = \text{Valid}|\mathrm{Ver}_{(i,0)}(x,\sigma_r) = \text{True}] \\ &\geq 1-\epsilon. \end{aligned} \tag{11}$$

If the protocol is ϵ'-secure against repudiation it must hold that $\Pr(rep) \leq \epsilon'$, which, using Eqs. (10) and (11) gives us

$$\begin{aligned}\epsilon' \geq \Pr(rep) &\geq (1-\epsilon)\Pr[\mathrm{Ver}_{(i,0)}(x,\sigma_r) = \text{True}] \\ &\geq (1-\epsilon)\frac{|S_i|}{|\Sigma|} \\ \Rightarrow \frac{|\mathrm{Ver}_{(i,0)}|}{|\Sigma|} &\leq \frac{\epsilon'}{1-\epsilon},\end{aligned}$$

where we have used the fact that $\Pr[\mathrm{Ver}_{(i,0)}(x,\sigma_r) = \text{True}] = \frac{|S_i|}{|\Sigma|}$. □

The size of the sets that pass the verification functions at different levels also plays an important role in permitting transferability. In fact, for a special class of USS protocols, such as the QS of Refs. [2, 18, 19], it is possible to provide conditions for these sets in order to achieve transferability and security against repudiation. These protocols, which we call *bit-mismatch* protocols, have the following properties. The set of possible signatures Σ is the set of all binary strings of n bits, i.e. $\Sigma = \{0,1\}^K$. For each possible message x, recipient P_i is given a random subset of positions p_i^x of size K of the integers $\{1,2,\ldots,n\}$. The recipient also receives verification bits v_i^x. Upon receiving a signature σ, a recipient collects the bits of σ at the positions corresponding to p_i^x to form a shorter string which we call σ_i. The verification functions are then given by

$$\mathrm{Ver}_{(i,l)}(x,\sigma) = \begin{cases} \text{True} & \text{if } h(\sigma_i, v_i^x) \leq s_l K \\ \text{False} & \text{otherwise} \end{cases} \tag{12}$$

for some $s_l \in [0, \frac{1}{2})$, which depends on the verification level l, and where $h(v_i^x, \sigma_i)$ is the Hamming distance between v_i^x and σ_i.

Observation 7 *For any correct bit-mismatch protocol which is transferable and secure against repudiation, with a majority-vote dispute resolution method, it must hold that $s_l < s_{l-1}$ for all l.*

Proof. Consider a cheating strategy by the signer in which she randomly flips each bit of the authentic signature $\text{Sign}(x)$ with probability p, leading to an altered signature σ'. For each participant, the choice of p induces a corresponding probability $q_{i,l}(p)$ that the altered signature will pass their verification function at level l. Since the protocol is correct, authentic signatures pass the verification functions of all participants at all levels, which implies that $q_{i,l}(0) = 1$ and $q_{i,l}(1) = 0$ for all l. The induced probability $q_{i,l}(p)$ is a continuous function of p[†], which implies that there must exist a value p_l^* such that, for some non-negligible $\delta > 0$, it holds that

$$1/2 - \delta < q_{i,l}(p_l^*) < 1/2 \tag{13}$$

for all participants P_i.

Now consider the case $l = 0$ and assume that $s_0 \leq s_{-1}$. By choosing p_0^* for her cheating strategy, the signer can create a signature which a given participant accepts with a non-negligible probability greater than $1/2 - \delta$ and smaller than $1/2$, according to Eq. (13). Moreover, since $s_0 \leq s_{-1}$, Eq. (13) implies that the probability that any other participant accepts the signature at level $l = -1$ must be smaller than $1/2$. In that case, with non-negligible probability, the majority of participants will reject the signature during dispute resolution, where they check the signature at level $l = -1$. Therefore, such a protocol cannot be secure against repudiation.

Similarly, for the case $l > 0$, a dishonest signer can choose p_l^* for her cheating strategy and have any given participant accept a signature at this level with probability at least $1/2 - \delta$. If $s_l \leq s_{l-1}$, when the participant who accepts the signature at level l transfers it to another person, the new participant will reject the signature at level $l - 1$ with non-negligible probability greater than $1/2$. Thus, such a protocol cannot offer transferability. □

Intuitively, the above proof states that the size of the set of signatures that pass the verification functions at a given level must increase for lower verification levels. This is illustrated in Fig. 3.

In the next section, we will examine previous QS protocols in light of our security framework. This will help illustrate our results with concrete examples as well as to showcase the importance of having a rigorous framework.

Here we briefly mention how previous three-party quantum signature protocols fit in our security framework. In particular, we consider the protocol DWA of Ref. [18] and the first protocol P1-WDKA from Ref. [2]. The experimental realisation in [21] is a variant of the DWA protocol. These two protocols do not require a quantum memory, and thus can be readily compared with classical USS schemes.

In these protocols, we have three participants given by the set $\mathcal{P} = \{P_0, P_1, P_2\}$. The set of possible messages is $X = \{0, 1\}$, i.e. we are interested in signing single-bit messages. There

[†]This probability distribution can be shown to be equal to the sum of two cumulative binomial distributions, which are continuous functions.

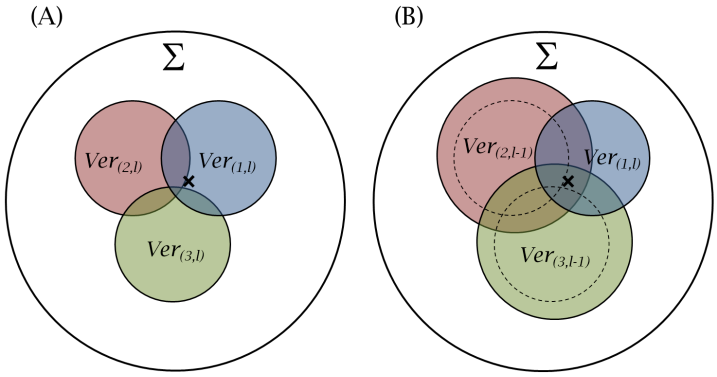

Fig. 3. For a given verification level l, a signer may produce a signature which, with non-negligible probability, passes the verification function at level l of participant P_1, but not of the other two participants at this same level. Such a signature is illustrated by a cross in the figure. Since more signatures are accepted at lower levels, when the other participants verify that same signature at level $l-1$, it now passes the verification function of all participants. This feature prevents repudiation and permits transferability.

is at most one dishonest participant. In the distribution stage, quantum states are exchanged and measured. At the end of this stage, the participants have obtained their verification algorithms. The set of possible signatures for the DWA protocol is $\Sigma = \{0,1\}^K$, while for P1-WDKA it is $\Sigma = \{0,1,2,3\}^K$, where K is the total length of the signature.

3.1 DWA protocol [18]

1. All the participants are connected by authenticated quantum channels and authenticated classical channels. The assumption of authenticated quantum channels means that the quantum messages which are transmitted are not altered during transmission. This can be guaranteed following a procedure similar to the parameter estimation phase of QKD [24].

2. For each message $x \in \{0,1\}$, the signer P_0 selects a string of bits σ^x, uniformly at random. For each 0 in the string σ^x he prepares the coherent state $|\alpha\rangle$ and for each 1 he prepares the coherent state $|-\alpha\rangle$. He then generates this sequence of coherent states twice and sends one copy of this sequence to P_1 and the other to P_2.

3. The recipients P_1, P_2 take their copies and pass them through an optical multiport (see [18] for details). The effect of this is the following. If all parties are honest, then they end up with the state P_0 sent, while if there was any deviation on P_0's side – for example P_0 sending different quantum states to P_1 and P_2 – then they end up with a symmetrised quantum state that is identical for both. This step is done to guarantee that the protocol is secure against repudiation.

4. Finally, each of the recipients measures the received sequence of coherent states $\bigotimes_k \left|(-1)^{\sigma^x_k}\alpha\right\rangle$ using unambiguous state discrimination [25, 26, 27]. The result is that each of the recip-

ients knows the correct bit value for the positions in which he obtains an unambiguous outcome. For participant i, we denote the bit string of outcomes as v_i^x and the positions for which they obtain unambiguous outcomes as p_i^x. The recipients have partial knowledge of the signature, but the sender does not know which bits are known to whom, and therefore he will not be able to repudiate.

5. Each participant P_i to verify that the signature $\tilde{\sigma}^x$ corresponds to the message x defines the verification function as follows. First, they form a shorter string $\tilde{\sigma}_i^x$ from $\tilde{\sigma}^x$ by keeping only the bits corresponding to the positions p_i^x for which they obtain unambiguous outcomes. The verification function of level l is then defined as

$$\mathrm{Ver}_{(i,l)}(x,\tilde{\sigma}) = \begin{cases} \text{True} & \text{if } h(\tilde{\sigma}_i^x, v_i^x) < s_l K \\ \text{False} & \text{otherwise} \end{cases} \tag{14}$$

where $h(\tilde{\sigma}_i^x, v_i^x)$ is the Hamming distance between the tested $\tilde{\sigma}_i^x$ and v_i^x, and s_l is a fraction defined by the protocol. Therefore, this protocol is a bit-mismatch protocol, as defined in the previous section. In the original protocol, there were only two thresholds s_a and s_v; the first was used to verify whether a signature is transferable and the second to verify just the origin of the signature. In our notation, $s_a = s_1$ and $s_v = s_0$. These fractions satisfy $s_0 > s_1$.

6. The signature function is given by $\mathrm{Sign}(x) = \sigma^x$.

7. Dispute resolution was not explicitly defined. However, it was implicit that a majority vote was to be used.

Remarks about the security of this protocol will be made after we give the description of the second protocol, since they have several similarities.

3.2 P1-WDKA protocol [2]

1. All the participants are connected by authenticated quantum channels and authenticated classical channels.

2. For each message $x \in \{0,1\}$, the signer P_0 selects a string σ^x of numbers from $\{0,1,2,3\}$, uniformly at random. For 0 he prepares the qubit state $|0\rangle$, for 1 the state $|+\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle)$, for 2 the state $|1\rangle$ and for 3 the state $|-\rangle = 1/\sqrt{2}(|0\rangle - |1\rangle)$. These are usually referred to as the BB84 states. He then generates this sequence of BB84 states twice and sends one copy of this string to P_1 and the second copy to P_2.

3. For each qubit he receives, recipient P_1 (P_2) randomly chooses whether to keep this state or forward it to P_2 (P_1). The effect of this process is that each of the qubits which P_0 sends may end up with either of the recipients. In other words, they have now symmetrized the quantum states they have, even if the sender P_0 initially deviated and sent different signatures σ^x to each one of them. For each message x, participant P_i defines the set $p_i^x \subset \{1,2,\ldots,K\}$ of positions for which they have a qubit.

4. Each of the recipients measures the received BB84 qubits using unambiguous state elimination [28, 29]. With this measurement, they *never* learn what state P_0 sent –

they only rule out one of the possible states. Therefore, for each position in p_i^x for which they had a qubit, they obtain a set of at most two states that are ruled out. The set of states they did not rule out is the set of allowed states, denoted by $A_j^x \subset \{0,1,2,3\}$ for each position j in p_i^x. Note that each A_j^x has at least two allowed states but not more than three. We then define the set of i-perfect signatures V_i^x as the set that contains all strings of symbols v_i^x where the value of the string for each of the positions in p_i^x is in the set A_j^x. Again, the recipients have partial knowledge of the signature σ^x and the sender P_0 is not aware of exactly what this knowledge is or for which positions this information was obtained.

5. Each participant P_i to verify that the signature $\tilde{\sigma}^x$ corresponds to the message x defines the verification function as follows. First, they form a shorter string $\tilde{\sigma}_i^x$ from $\tilde{\sigma}^x$ by keeping only the bits corresponding to the positions p_i^x for which they received qubits and thus have unambiguously ruled out states. The verification function for level l is then defined as

$$\mathrm{Ver}_{(i,l)}(x,\tilde{\sigma}) = \begin{cases} \text{True} & \text{if } \min_{v_i^x \in V_i^x} h(\tilde{\sigma}_i^x, v_i^x) < s_l K \\ \text{False} & \text{otherwise,} \end{cases} \tag{15}$$

where $h(\tilde{\sigma}_i^x, v_i^x)$ is the Hamming distance between $\tilde{\sigma}_i^x$ and v_i^x, and the minimum is taken over all i-perfect signatures. The fraction s_l is again defined by the protocol. In the original protocol, there were two thresholds s_a and s_v; the first was used to verify whether a signature is transferable and the second to verify just the origin of the signature. In our notation, $s_a = s_1$ and $s_v = s_0$, with $s_0 > s_1$.

6. The signature function is given by $\mathrm{Sign}(x) = \sigma^x$.

7. Dispute resolution was not explicitly defined. However, it was implicit that a majority vote was to be used.

The full security analysis of these protocols can be found in the original references. However, we here make a few remarks. First, we see that in a certain sense, these protocols are easier to analyse than general multiparty QS protocols because there is at most one dishonest participant. This significantly simplifies proofs for non-repudiation and transferability because the sender P_0 cannot have colluding parties. As we will see in the multiparty protocol below, having the sender colluding with recipients can lead to having honest participants totally disagreeing on fractions of the signature, and extra care is needed to address such possibilities.

Second, these protocols have a property that places strict demands on the noise level and imperfections in an implementation. The recipients P_1 and P_2 receive the same sequence of quantum states. Since they hold a legitimate copy of the state received by the other recipient, they have partial information about the other participant's verification algorithm. This makes it harder to guard against forging, to the point that security is only possible for low levels of noise and experimental imperfections. The security analysis is also complicated by the fact that the optimal forging attack depends on the states sent, e.g. on the amplitude α of the coherent states.

In any case, the intuition behind the security of these protocols is still the same. Forging is not possible because in order to deceive a participant P_2, the other participant P_1 should correctly guess the bit value for at least a fraction $s_v = s_0$ of the positions in which P_2 obtained an unambiguous outcome. Participant P_1 can use her copy to make a best guess, but this guess is never perfect, while the unambiguous measurement gives a perfect result when it does give a result, and therefore a legitimate participant always has an advantage. Repudiation in the case of three parties is essentially the same as non-transferability, since the aim is to make one recipient accept at level $l = 1$ and the other reject at the lower level $l = 0$. The security against this is guaranteed by the fact that the two recipients symmetrize their records, and therefore, from the point of view of P_0, he cannot make the one accept a lower threshold and then the other reject a higher threshold.

Finally, it is worth noting that, at the time Refs. [18] and [2] were written, the security framework we gave here did not exist. Therefore, the concept of dispute resolution and of the extra verification level $l = -1$ were not defined. Strictly speaking, for the full security of those protocols, we should define a new level $l = -1$ with threshold s_{-1} that obeys the condition that $s_{-1} > s_0 > s_1$. This is another good example of how the framework introduced in this work can prove fruitful in making more accurate statements and even in improving existing protocols.

In the next section, we use the security framework and properties developed so far to generalize the protocol P2-WDKA introduced in Ref. [2] to the case of many participants. We provide a full security proof against forging, repudiation and non-transferability.

4 Generalized P2-WDKA Protocol

In this protocol, which is a generalization of the protocol P2 of Ref. [2], we have $N + 1$ participants given by the set $\mathcal{P} = \{P_0, \cdots, P_N\}$. The set of possible messages is $X = \{x_1, \ldots, x_M\}$, where there are M different possible messages. Additionally, $\Sigma = \{0, 1\}^K$ is the set of possible signatures, and $K = nN$ is the length of the total signature, where n is an integer that depends on the required security parameters and is divisible by N.

As in any cryptographic protocol, we will make some trust assumptions. In particular, we assume that the number of honest participants[‡] is at least h. We can then define the fraction of dishonest participants as $d_f = 1 - h/N$. The maximum verification level $l_{\max}$ is determined by the allowed fraction of dishonest participants,

$$(l_{\max} + 1)d_f < 1/2. \tag{16}$$

The reason for this restriction will become clear later. The distribution stage of the protocol, which gives rise to the generation algorithm, proceeds as follows:

1. All the participants use quantum key distribution links in order to establish pairwise secret keys. Each recipient needs to share a secret key of nM bits with the signer P_0 and a secret key of $2\frac{nM}{N}(1 + \lceil \log_2 n \rceil)$ bits with each of the other recipients.

2. For each possible message $x \in X$, the signer selects a string σ^x of $K = nN$ bits uniformly at random and divides it into N sections $\{\sigma_1^x, \sigma_2^x, \ldots, \sigma_N^x\}$. The signer sends σ_i^x to participant P_i over a secure channel using their shared secret keys.

[‡]We assume that the adversaries are static, i.e. the participants are either honest or dishonest for the entire duration of the protocol.

3. For every possible message, each recipient randomly divides the set $\{1, 2, \ldots, n\}$ into N disjoint subsets $\{p^x_{i,1}, p^x_{i,2}, \ldots, p^x_{i,N}\}$ and uses the bit values of σ^x_i at the randomly chosen positions $p^x_{i,j}$ to form the string $v^x_{i,j}$.

4. For all $i \neq j$, each participant P_i transmits the string $v^x_{i,j}$ and the positions $p^x_{i,j}$ to participant P_j over a secure channel using their shared secret keys. Participant P_i keeps $v^x_{i,i}$ and $p_{i,i}$ to herself.

5. Each participant P_j defines a test for a section $\tilde{\sigma}^x_i$ of the signature to be verified as follows. First, they form a shorter string $\tilde{\sigma}^x_{i,j}$ from $\tilde{\sigma}^x_i$ by keeping only the bits corresponding to the positions $p^x_{i,j}$. The test is then defined as

$$T^x_{i,j,l}(\tilde{\sigma}^x_i) = \begin{cases} 1 & \text{if } h(\tilde{\sigma}^x_{i,j}, v^x_{i,j}) < s_l \frac{n}{N} \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

   where $h(\tilde{\sigma}^x_{i,j}, v^x_{i,j})$ is the Hamming distance between $\tilde{\sigma}^x_{i,j}$ and $v^x_{i,j}$ and s_l is a fraction defined by the protocol. These fractions satisfy

$$\frac{1}{2} > s_{-1} > s_0 > s_1 > \cdots > s_{l_{\max}}. \tag{18}$$

6. The verification function is defined as

$$\mathrm{Ver}_{(i,l)}(x, \tilde{\sigma}) = \begin{cases} \text{True} & \text{if } \sum_{j=1}^{n} T^x_{j,i,l}(\tilde{\sigma}^x_j) > N f_l \\ \text{False} & \text{otherwise} \end{cases} \tag{19}$$

   where f_l is a threshold fraction given by

$$f_l = \frac{1}{2} + (l+1) d_f. \tag{20}$$

7. The signature function is given by $\mathrm{Sign}(x) = \sigma^x$.

8. Majority vote is the dispute resolution method.

The main steps of the distribution stage are illustrated in Fig. 4.

The verification function, in words, accepts at level l if there are more than a fraction f_l of the sections $\{\tilde{\sigma}^x_1, \tilde{\sigma}^x_2, \ldots, \tilde{\sigma}^x_N\}$ that pass the test of the ith participant. This choice of the fraction f_l is made in order to satisfy a few constraints. First, we need the protocol for $l = -1$ to still require more than half of the tests to succeed, i.e. $f_{-1} > 1/2$. Second, we want the difference of the thresholds between two levels to exceed the fraction of dishonest participants i.e. $f_l - f_{l-1} > d_f$. Finally, by noting that $f_l \leq 1$ for all l, we determine the maximum value that l can take and this results in Eq. (16).

In the protocol, there are two different types of thresholds, s_l and f_l, both depending on the verification level l. The first threshold, s_l, determines whether a given part of the signature passes the test or not, by checking the number of mismatches at this part. The second threshold, f_l, determines how many parts of the signature need to pass the test in order for the signature to be accepted at that level.

An example of why different fractions for each verification level are needed is given by the following. Assume that one recipient, for example P_1, is a "spy" of an adversarial sender

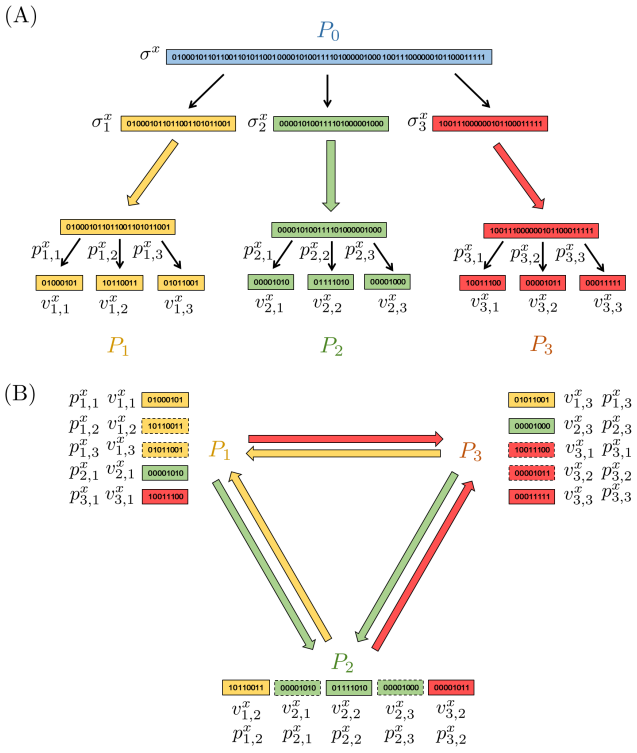

Fig. 4. Illustration of the protocol with four participants. In part (A), the sender divides a randomly generated string σ^x into three sections $\sigma_1^x, \sigma_2^x, \sigma_3^x$ and sends each of them to the corresponding participant over a secure channel, using a secret key previously generated using quantum key distribution. The secret channels are represented by thicker coloured arrows. The other participants divide the sections they receive to produce the strings $v_{i,j}^x$ alongside the corresponding positions $p_{i,j}^x$. In (B), the participants exchange the sections $v_{i,j}^x$ of the signature and the positions $p_{i,j}^x$ over secure channels. In the end, every participant keeps their original sections plus one additional section from each of the other participants, which they use for their verification functions. The sections in dashed boxes are known by the corresponding participant but are not used in the verification functions.

P_0, i.e. colludes with her in order to make two honest recipients P_2 and P_3 disagree on the validity of a signature. The spy can tell the sender the elements $(v^x_{1,2}, p_{1,2})$ and $(v^x_{1,3}, p_{1,3})$. The sender can then use this information to send a signature σ' that differs from the ideal signature σ only by flipping all the bit values at the positions determined by $p_{1,3}$. Recipient P_2 would accept the message, since he finds no errors. However, P_3 will find that *all* the bits of $v_{1,3}$ wrong, which will make his test fail. In general, if $d_f n$ dishonest participants exist, and if all of them are spies, two honest participants can differ by at most $d_f n$ tests. From Eq. (20), choosing $f_l - f_{l-1} = d_f$ allows the protocol to remain secure against this type of attack.

Finally, note that the important information defining the verification functions can be encoded in an $n \times n$ matrix, which we call the verification matrix. Each element of this matrix is a collection of M pairs of strings $(v^x_{i,j}, p^x_{i,j})$. The strings $v^x_{i,j}$ have length of $\frac{n}{N}$ bits, while the position records $p^x_{i,j}$ have length $\frac{n}{N} \times \lceil \log_2 n \rceil$ bits. Note that in $v^x_{i,j}$ and $p^x_{i,j}$, the first index corresponds to the section σ^x_i received by participant P_i, while the second index determines the other participant P_j with whom this string is shared. Importantly, it is not mandatory that these verification functions are constructed following the same steps as in the distribution stage outlined above. The security of the protocol relies only on the properties of the verification matrix and the value of other protocol parameters, which in principle may be generated by other means e.g. with the help of a trusted arbiter.

If the participants were honest during the above distribution stage, we end up exactly with the outcome of the ideal generation algorithm, which gives rise to the desired verification and signing functions. The important thing to notice is that deviating in the distribution stage is equivalent to being honest at this stage, but deviating at a later stage of the protocol. The sender gains nothing by sending a different signature to the recipients during the distribution stage, since this is equivalent to sending the correct signature during the distribution stage, but then sending a different signature at a later stage. The same holds for an adversarial recipient who is in coalition with the sender. On the other hand, an adversarial recipient P_i who wishes to forge a message by deviating and giving different $(v^x_{i,j}, p^x_{i,j})$, is not improving his chances to forge, since in order to forge a signature for participant P_1 for example, he will have to guess correctly the $(v^x_{i,1}, p^x_{i,1})$ and even if he is honest, he knows the $(v^x_{1,2}, p^x_{1,2})$.

We now proceed to prove the security of this protocol. In the following, for simplicity, we will drop the superscript labelling the message x from $v^x_{i,j}$, $p^x_{i,j}$ and $T^x_{(i,j,l)}$, and we will refer to participants by their index only, i.e. as i instead of P_i.

4.1 Security proofs

We will separately address the security of this protocol against forging, repudiation and non-transferability. We begin by noticing that the value of n must be chosen depending on other parameters and on the level of security. In particular, we want the probabilities for forging, non-transferability, and repudiation to decrease exponentially fast with n. However, the number of participants N also enters the security expressions. To make sure that the all cheating probabilities go to zero even when the number of participants is very large, in general we require that

$$n \geq \alpha N^{1+\delta}, \tag{21}$$

where $\alpha \gg 1$ is a large positive constant and δ a small positive constant.

Forging. In order to forge, a coalition C which does not include the signer needs to output a message-signature pair $(x, \tilde{\sigma})$ that is i-acceptable for some $i \notin C$. In general, according to our definitions, we consider forging successful if the coalition can deceive *any* honest participant, and not a fixed one. Here, for simplicity, we restrict attention to trying to deceive a fixed participant, and we will prove that this probability decays exponentially fast with the parameter n. At the end, we will extend this to the general case where the target is not a fixed participant. Therefore, for now, we fix the recipient that the coalition wants to deceive to be simply i.

Recall that a signature $\tilde{\sigma}$ is i-acceptable if $\text{Ver}_{(i,0)}(x, \tilde{\sigma}) = \text{True}$. By the definition of the verification functions of our protocol, this means that the coalition should output a signature $\tilde{\sigma}$ such that participant i accepts Nf_0 tests at level zero, $T_{j,i,0}(\tilde{\sigma}^j)$. From Eq. (20), we have that $f_0 = \frac{1}{2} + d_f$. By the definition of the protocol, the number of members in a coalition is at most Nd_f. The coalition knows the pairs $(v_{j,i}, p_{j,i})$ for all $j \in C$, so they can use this knowledge to trivially pass Nd_f tests. It follows that in order to forge, the coalition must pass at least $N(f_0 - d_f) = \frac{N}{2}$ tests out of the $N(1 - d_f)$ tests that they do not have access to. The first step to compute the probability that they can do this is to calculate the probability of passing a single test $T_{j,i,0}$ for $j \notin C$.

1. We denote the probability to pass a test at level $l = 0$ for a coalition with no access to the pair $(v_{i,j}, p_{i,j})$ by p_t. Because the strings $(v_{i,j}, p_{i,j})$ were transferred over secure channels by honest recipients, they are completely unknown to the coalition and hence the probability of guessing correctly a single bit of $v_{i,j}$ is exactly $1/2$. In order to pass the test, the coalition needs to guess at least a fraction s_0 of bits out of a total of $\frac{n}{N}$ bits. The probability that they can achieve this can be bounded using Hoeffding's inequality as

$$p_t \leq \exp\left(-2(1/2 - s_0)^2 \frac{n}{N}\right), \tag{22}$$

   which decays exponentially with the number $\frac{n}{N}$ provided that $s_0 < 1/2$. Note that, from by Eq. (21), we know that this term decays exponentially even for $N \to \infty$.

2. Now we will give a bound for the probability of forging against a fixed participant. This can be obtained by computing the probability of passing at least one of the unknown $N(1 - d_f)$ tests, which is given by

$$\begin{aligned} \Pr(\text{FixedForge}) &< 1 - (1 - p_t)^{N(1-d_f)} \approx N(1 - d_f)p_t \\ &\leq (1 - d_f)N \exp\left(-2(1/2 - s_0)^2 \frac{n}{N}\right), \end{aligned} \tag{23}$$

   where we have used the fact that $p_t \ll 1$ in the approximation. Again, this probability goes to zero exponentially fast in the parameter n. Note also that, by Eq. (21), this expression goes to zero even for the case $N \to \infty$, as the term with p_t goes exponentially fast to zero while the other term grows only linearly in N.

3. We have now computed the probability to deceive a fixed participant i. The total number of honest participants is $N(1 - d_f)$ and for successful forging we require that any one of them is deceived. We therefore obtain

$$\Pr(\text{Forge}) = 1 - (1 - \Pr(\text{FixedForge}))^{N(1-d_f)} \lesssim N^2(1-d_f)^2 \exp\left(-2(1/2 - s_0)^2 \frac{n}{N}\right). \quad (24)$$

Transferability. In order to break the transferability of the protocol, a coalition C which includes the signer P_0 must generate a signature that is accepted by recipient $i \notin C$ at level l, while rejected by another recipient $j \notin C$ at a level $l' < l$. To provide an upper bound, we allow for the biggest coalition C that includes Nd_f participants, i.e. all the dishonest participants. For simplicity, again we will fix the participants whom the coalition is trying to deceive to be the ith and jth, while all the other honest participants are labelled with the index k. In general, according to our definitions, transferability fails if the coalition forms a signature that is not transferable for *at least one* pair of honest participants i, j. Therefore, we should take into account all possible pairs of honest participants. Here, we first focus on the case of a fixed pair of participants, and we give at the end the more general expressions. The members of the coalition C are labelled with the index c.

We first give a sketch of the proof. The first step is to compute $p_{m_{l,l'}}$. This is the probability that the tests corresponding to a part of the signature σ^k of an honest recipient k satisfy the following conditions: (i) The test $T_{k,i,l}$ of an honest recipient i at level l is passed *and* (ii) the test $T_{k,j,l'}$ of another honest recipient j at a level $l' < l$ is failed. The second step of the proof is to prove that in order for non-transferability to be successful, there must be at least one test corresponding to an honest participant which the two recipients i and j disagree on. The third step is to combine the previous two steps to provide a bound for the probability of non-transferability for a fixed pair of recipients. Finally, we can use the previous results to bound the probability of non-transferability for any pair of honest recipients.

1. First, we compute $p_{m_{l,l'}}$, which is the probability that the kth test $T^x_{k,i,l}$ of an honest recipient i at level l is accepted *and* the test $T^x_{k,j,l'}$ of another honest recipient j at a level $l' < l$ is rejected. The relevant part of the signature is σ^k, where k is an honest recipient. The two parts of the verification matrix that are relevant are $(v_{k,i}, p_{k,i})$ and $(v_{k,j}, p_{k,j})$. Since the sender is in the coalition, they know the values of all the sections $v_{i,j}$, but they are completely ignorant of the positions $p_{k,i}$ and $p_{k,j}$, since participants k, i and j are all honest. The coalition can decide to send signatures in such a way that they introduce an average fraction of mistakes p_e compared to the ideal signature that was used to generate the verification algorithms. Thus, the average fraction of mistakes is under their control. Since the protocol is symmetric for all participants, this average fraction of mistakes will be the same for all honest participants and in particular for both i and j.

   To compute a bound on the joint probability of i accepting at level l and j rejecting at level $l-1$ we will consider

$$\begin{aligned} p_{m_{l,l'}} &= \Pr\left(i \text{ accepts at level } l \text{ AND } j \text{ rejects at level } l'\right) \\ &\leq \min\{\Pr\left(i \text{ accepts at level } l\right), \Pr\left(j \text{ rejects at level } l'\right)\}. \end{aligned} \quad (25)$$

   The probability of passing the test at level l with an average error p_e can be bounded using Hoeffding's inequalities to be below $\exp\left(-2(p_e - s_l)^2 \frac{n}{N}\right)$. This is the case since

the expected number of mistakes are $\frac{n}{N}p_e$ while the mistakes that are tolerated for acceptance are $\frac{n}{N}s_l$. We note that this expression holds for $p_e > s_l$. However, as we will see, for $p_e < s_l$ the probability for the participant rejecting at level $l' < l$ will be even smaller, and since for our bound we consider the minimum of those two probabilities, we can assume that $p_e > s_l$.

The probability of failing the test at level l' with average errors p_e can similarly be bounded to be smaller than $\exp\left(-2(s_{l'} - p_e)^2 \frac{n}{N}\right)$. This is since the expected mistakes are $\frac{n}{N}p_e$ while the mistakes needed to fail are more than $s_{l'}\frac{n}{N}$. We note, that this expression holds for $p_e < s_{l'}$. However, as we have seen, for $p_e > s_{l'}$, the probability for the participant accepting at level l will be even smaller (recall $s_l < s_{l'}$), and since for our bound we consider the minimum of those two probabilities, we can assume that $p_e < s_{l'}$.

Therefore, the coalition must choose a value of p_e satisfying

$$s_l < p_e < s_{l-1}. \tag{26}$$

Since we are taking the minimum over both cases, the best choice for the coalition is to have both probabilities coincide. This is achieved by using a fraction of errors $p_e = (s_l + s_{l-1})/2$ and in that case we obtain the bound

$$p_{m_{l,l'}} \leq \exp\left(-\frac{(s_{l'} - s_l)^2}{2}\frac{n}{N}\right) \tag{27}$$

which decays exponentially with $\frac{n}{N}$ and it also depends on the difference $(s_{l'} - s_l)$.

2. It is trivial for the coalition to make two recipients disagree in any way they wish for the results of a test that involves a member of the coalition, i.e. they can make $T^x_{c,i,l}$ and $T^x_{c,j,l'}$ take any values they wish. However, the number of those tests are at most Nd_f, which is the maximum number of members in the coalition.

   For the participant i to accept a message at level l, he needs a fraction greater than f_l of the tests to pass at this level. On the other hand, for the participant j to reject the message at level l', a fraction greater than $1 - f'_l$ of tests must fail at this level. Therefore, even taking the best case for the coalition, which is $l' = l - 1$, since it holds that $f_l = f_{l-1} + d_f$, in order for the non-transferability to be successful, the honest participants i and j need to disagree on at least $Nd_f + 1$ tests. As we saw, the coalition can easily make them disagree on the Nd_f tests originating from them, but the participants i and j still have to disagree on at least one more test originating from an honest participant.

3. In order for the coalition to successfully cheat, the number of tests that pass for the ith recipient must be at least $Nf_l + 1$. Out of those tests we can assume that Nd_f were due to the coalition, but there are still $N(f_l - d_f) + 1$ tests that the coalition does not have access to. In order for the non-transferability to be successful, at least one of these $N(f_l - d_f) + 1$ tests should fail for participant j at level $l' = l - 1$. The probability that they agree in all of them is $(1 - p_{m_{l,l'}})^{N(f_l - d_f)+1}$ and therefore the probability for fixed

non-transferability can be bounded as

$$\begin{aligned}\Pr(\text{FixedNonTrans}) &\leq 1-(1-p_{m_{l,l'}})^{N(f_l-d_f)+1}\\ &\approx [N(f_l-d_f)+1]\,p_{m_{l,l'}}\\ &\leq [N(f_l-d_f)+1]\exp\left(-\frac{(s_{l'}-s_l)^2}{2}\frac{n}{N}\right)+O(p^2_{m_{l,l'}}).\end{aligned} \tag{28}$$

This goes to zero exponentially with $\frac{n}{N}$. Note that the first term scales linearly in N, but $p_{m_{l,l'}}$ decays exponentially with $\frac{n}{N}$, therefore with the choice of Eq. (21) this probability also vanishes at all limits of interest.

4. Finally, we should consider the general case, where the participants i, j are not fixed. Again, we can see that because the probability for fixed parties decays exponentially in the parameter n, the protocol remains secure. The number of honest pairs of participants is $[N(1-d_f)][N(1-d_f)-1]/2 := N_p$, so we obtain

$$\Pr(\text{NonTrans}) = 1-(1-\Pr(\text{FixedNonTrans}))^{N_p} \approx O(N^3)\exp\left(-\frac{(s_{l'}-s_l)^2}{2}\frac{n}{N}\right). \tag{29}$$

Non-repudiation. In order to repudiate, a coalition C including the sender P_0 generates an i-acceptable signature for some $i \notin C$, where invoking the dispute resolution DR results in Invalid. This means that the coalition wants to make any participant accept a signature at level $l = 0$, but then have the majority of participants to reject the same signature at level $l = -1$. We can actually reduce this problem to the special case of non-transferability from level $l = 0$ to level $l = -1$ in the following three steps.

1. We first find the probability of non-transferability for a fixed pair of participants, i.e. from a fixed honest participant i at level $l = 0$ to another fixed honest participant j at level $l = -1$. We denote this probability by p_1 and, as found before, it can be bounded by

$$p_1 \lesssim [N(f_0-d_f)+1]\,p_{m_{0,-1}} \leq \left(\frac{N}{2}+1\right)\exp\left(-\frac{(s_{-1}-s_0)^2}{2}\frac{n}{N}\right), \tag{30}$$

where we have used the fact that $(f_0 - d_f) = \frac{1}{2}$ from Eq. (20).

2. The second step is to note the following. For a fixed recipient i to accept at $l = 0$, it means that at least $Nf_0 + 1 = N(\frac{1}{2} + d_f) + 1$ parts of his signature were accepted. Out of these, $\frac{N}{2} + 1$ must have come from honest participants. Now, each of those honest participants that sent i a part that passed his tests also sent the other honest participants sections which, with probability $1-p_1$, pass their tests at level $l = -1$. For a message to be declared invalid in the dispute resolution DR, half of the participants have to reject. However, at least $\frac{N}{2}+1$ are unlikely to reject, since the probability that they do reject is p_1, which can be made arbitrarily small. In other words, for the DR to give Invalid, at least one of the honest participants needs to fail the transferability for a fixed pair of participants.

3. It is now clear that if no fixed pair of honest participants i, j fails the transferability for levels $l = 0$ to $l = -1$, then the coalition cannot repudiate. This leads to the following bound for the probability of repudiation,

$$\begin{aligned} \Pr(\text{Rep}) &\leq 1 - (1 - p_1)^{N_p} \approx N_p p_1 + O(p_1^2) \\ &\leq O(N^3) \exp\left(-\frac{(s_{-1} - s_0)^2}{2} \frac{n}{N} \right), \end{aligned} \tag{31}$$

where N_p as before is the number of honest pairs $[N(1-d_f)][N(1-d_f)-1]/2$ and p_1 decays exponentially with $\frac{n}{N}$.

We have seen that all security parameters, from Eqs. (23), (28) and (31), go to zero exponentially fast with $\frac{n}{N}$, provided correct choices of s_l and f_l are made. As stressed before, by Eq. (21), we also know that these parameters go to zero even if the number of participants N goes to infinity.

Secure channels from QKD. Security proofs for quantum key distribution (QKD) rely on the assumption that the parties wishing to exchange a secret key behave honestly. In the context of our multiparty protocol for quantum signature schemes, this assumption does not hold, since some of the participants performing QKD may be dishonest. However, we can show that this does not present a problem for the security of our protocol in three steps. Similar arguments are made in [24].

Step 1: Only honest-dishonest QKD links may be affected. The first observation is that dishonest behaviour during QKD may only be an issue when the QKD link connects an honest participant with a dishonest one. For two honest participants, standard QKD security proofs apply, so we are not concerned with this scenario. For the case of two dishonest participants, since all members of the coalition have access to the same information – as is assumed in our security definitions – it is irrelevant whether they behave honestly during QKD. Similarly, honest participants do not eavesdrop on dishonest participants, so there are no consequences to the security of the QS protocol.

In the following two steps we will show that for the case of an honest and a dishonest participant using QKD to establish a shared secret key, *any* adversarial behaviour during the QKD stage of the protocol is equivalent to a dishonest behaviour in subsequent parts of the protocol. Therefore, we can assume that the participants were honest during the QKD stage and examine all possible deviations for later stages of the protocol.

Step 2: No-gain from leaking information. At the end of a QKD protocol, an honest participant P_i holds a key register X which, in the ideal case, is identical to the string Y of the other participant P_c and is completely unknown to any other party. This means that any dishonest behaviour by participant P_c can only lead to two possible outcomes: (i) The registers X and Y are not identical, or (ii) X is correlated with the register of another party. Since we assume that all dishonest participants are in coalition, all of them have perfect knowledge of the register Y, so there is no need to eavesdrop information about this string. They of course benefit from knowledge of X, but they can have perfect knowledge of X simply if P_c behaves honestly during QKD. Therefore, leakage of information does not help the adversarial coalition.

Step 3: No-gain from imperfect keys. Similarly, if there are mismatches between the registers X and Y, any message which is transmitted secretly by using a one-time pad with

either X or Y will be received with errors in all positions in which X and Y differ. However, if Y is used by P_c to transmit a message to the honest participant P_i, the situation is exactly equivalent to one in which they have identical secret keys, but P_c decided to introduce errors in the message sent to P_i. Similarly, if P_i is the one sending the message, the situation is equivalent to the keys being identical but participant P_c introducing errors after receiving the message. In fact, since in order to cheat, the coalition needs to know the verification function of the honest participants, their optimal strategy is to be honest during the QKD stage and have a perfect copy of the other participants' secret keys. Therefore, the security of QKD is only relevant in a quantum signature scheme in order to protect honest participants who want to establish a secret key. It is precisely in this regime that standard QKD proofs apply.

5 Discussion

In this work, we have provided a full security framework for quantum signature schemes. We have generalized the security definitions of Swanson and Stinson [1] to allow for quantum schemes and different levels of verification. Additionally, we have proven several properties that USS protocols, quantum or classical, must satisfy in order to achieve their security goals. Together, these results form a powerful set of tools to be employed in the understanding and development of improved protocols in a general setting.

In fact, we have done just that by using our security framework to generalize the P2-WDKA protocol of Wallden et. al [2] to the multiparty case. This protocol is secure against forging, repudiation, and non-transferability, relying on minimal security assumptions. Interestingly, the quantum-mechanical features responsible for the security of the protocol can be completely outsourced to quantum key distribution (QKD), where a vast literature of sophisticated security proofs already exists. This feature also addresses the issue of authentication in quantum signature schemes: we can simply use QKD to generate new secret keys to be used in the authentication of future instances of a signature protocol. Finally, since this protocol can be implemented using any point-to-point QKD network, it is already practical, making experimental demonstrations in the short-term future a real and exciting possibility.

As a consequence of our results and those of Ref. [2], the status of unconditionally secure signature schemes should be considered analogous to that of secure communication, where a classical protocol – the one time-pad – already exists and can guarantee information-theoretic security at the expense of shared secret keys. Quantum communication can then be used to establish these secret keys via unsecured quantum channels. Similarly, for signature schemes, there exist classical protocols – such as our generalized P2-WDKA protocol – which provide information-theoretic security at the expense of shared secret keys. Remarkably, even in the setting where parties are dishonest, quantum key distribution can be used to establish the secret keys. Overall, we can now understand unconditionally secure signature schemes as a practical application of quantum key distribution. Future work can focus on optimizing these classical protocols, for example in reducing the length of the secret keys that need to be exchanged as a function of the message size. Additionally, it is important to continue to study protocols where quantum communication can be used to construct quantum signature schemes without the need to distil a secret key. Those schemes could offer advantages in terms of scalability, or in terms of extending the distance between parties, and thus be proven more useful in this respect. For example, Ref. [24] discusses how such "direct quantum"

signature schemes may be practical even if the quantum bit error rate is too high to allow the distillation of a secure key.

Acknowledgements

The authors would like to thank A. Ignjatovic, N. Lütkenhaus and D. Stinson for valuable discussions. In particular, the authors would like to thank V. Dunjko for discussions in our previous collaboration, especially [2], where the three-party protocol P2-WDKA that was generalised in the present paper was invented, mainly by him. This work was supported by Industry Canada, the NSERC Strategic Project Grant (SPG) FREQUENCY, the NSCERC Discovery Program and the UK Engineering and Physical Sciences Research Council (EPSRC) under EP/K022717/1 and EP/M013472/1. J.M. Arrazola is grateful for the support of the Mike and Ophelia Lazaridis Fellowship. P.W. gratefully acknowledges support from the COST Action MP1006.

References

1. C. M. Swanson and D. R. Stinson, "Unconditionally secure signature schemes revisited," *Information Theoretic Security*, pp. 100–116, 2011.
2. P. Wallden, V. Dunjko, A. Kent, and E. Andersson, "Quantum digital signatures with quantum-key-distribution components," *Phys. Rev. A*, vol. 91, p. 042304, 2015.
3. A. M. Childs and W. Van Dam, "Quantum algorithms for algebraic problems," *Reviews of Modern Physics*, vol. 82, no. 1, p. 1, 2010.
4. D. Chaum and S. Roijakkers, "Unconditionally-secure digital signatures," *Advances in Cryptology*, pp. 206–214, 1991.
5. E. F. Brickell and D. R. Stinson, "Authentication codes with multiple arbiters," in *Advances in Cryptology*, pp. 51–55, Springer, 1988.
6. G. Hanaoka, J. Shikata, Y. Zheng, and H. Imai, "Unconditionally secure digital signature schemes admitting transferability," in *Advances in Cryptology*, pp. 130–142, Springer, 2000.
7. G. Hanaoka, J. Shikata, Y. Zheng, and H. Imai, "Efficient and unconditionally secure digital signatures and a security analysis of a multireceiver authentication code," in *Public Key Cryptography*, pp. 64–79, Springer, 2002.
8. T. Johansson, "On the construction of perfect authentication codes that permit arbitration," in *Advances in Cryptology*, pp. 343–354, Springer, 1994.
9. T. Johansson, "Further results on asymmetric authentication schemes," *Information and Computation*, vol. 151, no. 1, pp. 100–133, 1999.
10. R. Safavi-Naini, L. McAven, and M. Yung, "General group authentication codes and their relation to unconditionally-secure signatures," in *Public Key Cryptography*, pp. 231–247, Springer, 2004.
11. G. J. Simmons, "Message authentication with arbitration of transmitter/receiver disputes," in *Advances in Cryptology*, pp. 151–165, Springer, 1988.
12. G. J. Simmons, "A cartesian product construction for unconditionally secure authentication codes that permit arbitration," *Journal of Cryptology*, vol. 2, no. 2, pp. 77–104, 1990.
13. C. E. Shannon, "Communication theory of secrecy systems*," *Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.
14. U. M. Maurer, "Secret key agreement by public discussion from common information," *Information Theory, IEEE Transactions on*, vol. 39, no. 3, pp. 733–742, 1993.
15. D. Gottesman and I. Chuang, "Quantum digital signatures," *arXiv preprint quant-ph/0105032*, 2001.
16. J. Müller-Quade, "Quantum pseudosignatures," *Journal of Modern Optics*, vol. 49, no. 8, pp. 1269–1276, 2002.

17. X. Lu and D. Feng, "Quantum digital signature based on quantum one-way functions," in *Advanced Communication Technology, 2005, ICACT 2005. The 7th International Conference on*, vol. 1, pp. 514–517, IEEE, 2005.
18. V. Dunjko, P. Wallden, and E. Andersson, "Quantum digital signatures without quantum memory," *Physical Review Letters*, vol. 112, no. 4, p. 040502, 2014.
19. J. M. Arrazola and N. Lütkenhaus, "Quantum communication with coherent states and linear optics," *Physical Review A*, vol. 90, no. 4, p. 042335, 2014.
20. P. J. Clarke, R. J. Collins, V. Dunjko, E. Andersson, J. Jeffers, and G. S. Buller, "Experimental demonstration of quantum digital signatures using phase-encoded coherent states of light," *Nature communications*, vol. 3, p. 1174, 2012.
21. R. J. Collins, R. J. Donaldson, V. Dunjko, P. Wallden, P. J. Clarke, E. Andersson, J. Jeffers, and G. S. Buller, "Realization of quantum digital signatures without the requirement of quantum memory," *Phys. Rev. Lett.*, vol. 113, p. 040502, 2014.
22. R. Amiri and E. Andersson, "Quantum signatures," Entropy, vol. 17, pp 5635-5659, 2015.
23. J. L. Carter and M. N. Wegman, "Universal classes of hash functions," *J. Comp. Syst. Sci.*, vol. 18, pp. 143–154, 1979.
24. R. Amiri, P. Wallden, A. Kent, and E. Andersson, "Secure quantum signatures using insecure quantum channels," preprint arXiv:1507.02975, 2015.
25. I. Ivanovic, "How to differentiate between non-orthogonal states," *Physics Letters A*, vol. 123, no. 6, pp. 257 – 259, 1987.
26. D. Dieks, "Overlap and distinguishability of quantum states," *Physics Letters A*, vol. 126, no. 56, pp. 303 – 306, 1988.
27. A. Peres, "How to differentiate between non-orthogonal states," *Physics Letters A*, vol. 128, no. 12, pp. 19 –, 1988.
28. C. M. Caves, C. A. Fuchs, and R. Schack, "Conditions for compatibility of quantum-state assignments," *Phys. Rev. A*, vol. 66, p. 062111, 2002.
29. S. Bandyopadhyay, R. Jain, J. Oppenheim, and C. Perry, "Conclusive exclusion of quantum states," *Phys. Rev. A*, vol. 89, p. 022336, 2014.