Deniable Public-Key Authenticated Quantum Key Exchange

Jeroen van Wier, Arash Atashpendar, and Peter Roenne

Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg

Abstract. In this work, we explore the notion of deniability in public-key authenticated quantum key exchange (QKE), which allows two parties to establish a shared secret key without leaving any evidence that would bind a session to either party. The deniability property is expressed in terms of being able to simulate the transcripts of a protocol. The ability to deny a message or an action has applications ranging from secure messaging to secure e-voting and whistle-blowing. While quite well-established in classical cryptography, it remains largely unexplored in the quantum setting. Here, we first present a natural extension of classical definitions in the simulation paradigm to the setting of quantum computation and formalize the requirements for a deniable QKE scheme. We then prove that the BB84 variant of QKE, when authenticated using a strong designated verifier signature scheme, satisfies deniability and, finally, propose a concrete instantiation.

Keywords: Public-key Cryptography, Deniability, Quantum Cryptography, Post-Quantum Cryptography, Quantum Key Distribution, Authenticated Quantum Key Exchange, Designated-Verifier Signatures

1 Introduction

Among the wide variety of anticipated cybersecurity challenges, the possibility of the emergence of scalable quantum computers poses a serious threat to our current information security infrastructure and has been receiving increasingly more attention from the information security community over the past few decades. While quantum computing would have its advantages, Shor's algorithm [Sho94] for efficiently computing discrete logarithms and performing integer factorization showed that quantum computing is a double-edged sword as it can be equally damaging when used for the purpose of compromising public-key (PK) cryptosystems that guarantee the security of today's modern communication systems.

These concerns are perhaps best exemplified by recent advances that have prompted calls by the National Security Agency (NSA) for transitioning to post-quantum (PQ) secure cryptosystems and the call for PQ secure proposals, initiated by the National Institute of Standards and Technology (NIST) as part of a standardization process for post-quantum algorithms [SN17].

On the other hand, Quantum Key Exchange (QKE), provides security without relying on computational assumptions as in PQ key exchange protocols, but at the cost of developing new infrastructure to support quantum channels. Whereas such quantum communication infrastructures for a long time were mostly of academic interest, both terrestrial and satellite networks are now being deployed in practice and planned at large scale, see e.g. [Che+21, Eur].

Deniability constitutes a subtle and fundamental concept in cryptography that has many applications ranging from secure messaging (e.g., the Signal protocol) to coercion-resistance in secure e-voting to deniable transmission and storage in the context of data breaches. On a more fundamental and theoretical level, deniability shares an intimate connection with incoercible secure multi-party computation [CG96]. Yet, it has received very little attention in the quantum setting and thus presents a wealth of open questions.

Attempts at providing security against quantum adversaries can be broken down into two classes, namely those that largely rely on classical constructions that are conjectured to be quantum-secure, often classified as post-quantum cryptography, e.g., lattice-based cryptography, and those that make use of quantum information processing and thus fall in the realm of quantum cryptography, such as quantum key exchange. In both cases, and perhaps more surprisingly in the context of quantum cryptography, the notion of deniability has been largely neglected to the extent that there exist only a few works on this topic in the literature [Bea02, Ata+18, Ata19].

In this work, we focus on deniability in public-key authenticated QKE in the simulation paradigm wherein a scheme is considered to be deniable if its transcripts can be simulated. This becomes relevant in a setting with two parties A and B, in which one of the parties is dishonest (i.e., the adversary $\mathcal{M}$) and the goal is to prevent either one from proving to a judge that they exchanged a key with a specific party in a given session. Now, if the transcript obtained by $\mathcal{M}$ could have been simulated without having access to the honest party's secret key, the resulting evidence cannot convincingly associate a specific party with a given session. Note that in the case of deniable key exchange, not only the communication but also the resulting session secret should be simulatable [DRGK06].

The particular choice of considering public-key authentication for QKE is motivated by the following observations. As already pointed out in the seminal work of Di Raimondo et al. [DRGK06] on deniable authenticated key exchange for classical schemes, deniability for symmetric key exchange protocols in the simulation paradigm is trivially satisfied. Secondly, to cope with the criticism that unconditionally secure QKE requires pre-shared symmetric keys for authentication, a problem that scales quadratically with the number of connected users, the idea of using public-key authentication algorithms for performing QKE with everlasting security had been considered for quite a while until its security was formally proved by Mosca et al. in [MSU13]. For a detailed analysis of PK-authenticated QKE, we refer the reader to [IM11].

While PK-authenticated QKE solves the problem of pre-shared keys, the signatures also introduce non-repudiation. In this paper, we demonstrate how a deniable QKE can be constructed in the PK setting, in order to regain the deniability from the symmetric setting, by authenticating via quantum-safe strong designated verifier signatures (SDVS), e.g. obtained from lattices as in [NJ16], thus being potentially quantum secure. This implies that the resulting deniable QKE scheme would provide everlasting security, i.e., unless the adversary breaks the authentication during a limited window of attack, the derived shared secret key retains information theoretic security. Note that due to a unique property of QKE, namely that of non-attributability [IM11] (i.e., the final secret key being completely independent of the classical communication and the initial pre-shared key), the simulatability of the classical communication and that of the secret key itself can be considered separately. The latter follows from the inherent properties of QKE and, to establish the deniability of our solution, it thus suffices to show that the transcript of the authentication can be simulated, i.e., the authentication is deniable.

Related Work Compared to classical cryptography, deniability remains largely unexplored in the context of quantum and post-quantum cryptography. More specifically, in a paper by Beaver [Bea02] focusing on a setting motivated by an earlier work by Canetti et al. [Can+97] on deniable encryption, it is mainly argued that QKE protocols are not necessarily deniable. In a related work [Ata+18], Atashpendar et al. revisit Beaver's analysis and formalize the problem of coercer-deniability in terms of the indistinguishability of coercer views, which considers a scenario wherein the adversary can demand that the honest parties reveal their private randomness in order to verify whether or not their revealed secret key is real or fake. They also establish a link between covert quantum communication and deniability, as well as a relation between entanglement distillation and information theoretic deniability.

However, [Ata+18] concludes with a number of open questions, including an analysis of public-key authenticated QKE in the simulation paradigm, which is the focus of our work.

The work of Canetti et al. [Can+97] led to a long series of works on deniability for various cryptographic primitives, including a formalization of deniability for authenticated key exchange in the simulation paradigm by Di Raimondo et al. [DRGK06], which in turn was an extension of the definitional work of Dwork et al. [DNS04] on deniable authentication in the context of zero-knowledge proofs. We refer the reader to [Ata+18, Ata19] and references therein for more details on deniability in cryptography.

Contributions We adopt the security framework for authenticated QKE given in [MSU13] and adapt the classical definition of deniable AKEs [DRGK06] to the quantum setting for public-key authenticated QKE and formulate it in terms of the simulatability of protocol transcripts in a game-based setting.

We prove in Theorem 1 that a public-key authenticated QKE protocol satisfies deniability when authenticated using an SDVS with non-transferability

against quantum adversaries. We also propose the first concrete instance of a deniable PK-authenticated QKE, which is a BB84 variant whose deniability follows as a corollary of Theorem 1.

2 Preliminaries

Notation We write $y \leftarrow A(x)$ to denote that algorithm A outputs y on input x and use $\perp \leftarrow A$ to denote that A produced an error. We write $\boldsymbol{v}$ to denote a vector of values and v_i to denote the i-th value of this vector. We use $\kappa \in \mathbb{N}$ to denote the security parameter, and implicitly assume it is passed to all algorithms of schemes in unary, i.e. in the form 1^κ. Lastly, we use $f(n) \leq \mathrm{negl}(n)$ to denote that a function f is *negligible*, which means that $f(n)^{-1}$ is superpolynomial.

While we deal with notions from quantum computing, their understanding is not critical to the work and thus we refer to [Wat18] for an overview of quantum computing. We adopt the standard bra-ket notation from quantum computing. We denote pure states with $|\cdot\rangle$ and mixed states with ρ. We use $(+)$ to denote the $\{|0\rangle, |1\rangle\}$ basis and $(\times)$ to denote the $\{|+\rangle, |-\rangle\}$ basis. We denote the class of quantum polynomial-time algorithms as QPT (the quantum equivalent of PPT) and use $\mathcal{D}_1 \approx_q \mathcal{D}_2$ to denote that two probability distributions cannot be distinguished with more than negligible probability by any QPT distinguisher.

Strong Designated Verifier Signatures (SDVS) The classical communication in authenticated QKE poses a challenge for deniability because a receiver must be able to verify that a message came from the correct sender, but this task must be impossible for any eavesdropper. Note that we focus explicitly on the setting of public-key authentication, which presents the problem that a standard signature, verifiable by anyone with the signer's public key, would prove the involvement of the signer. In the symmetric-key setting, this problem would be trivially solved, as any signature can only be verified by the signer and the intended recipient, and either party can create the same signatures. To achieve these same properties in the public-key setting, we make use of strong designated verifier signatures.

Definition 1. *A* designated verifier signature scheme (DVS scheme) *is a tuple* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Simulate})$ *of* PPT *algorithms such that:*

- Setup*: Produces the public parameters of a scheme,* params*. It is implicitly assumed that these parameters are passed to the following algorithms.*
- KeyGen*: Produces a keypair* $(\mathsf{pk}, \mathsf{sk})$*.*
- $\mathsf{Sign}_{S\to V}(m) := \mathsf{Sign}(\mathsf{sk}_S, \mathsf{pk}_S, \mathsf{pk}_V, m)$*: Upon input of a sender's keypair, a verifier's public key, and a message* m*, produces a signature* σ*.*
- $\mathsf{Verify}_{S\to V}(m, \sigma) := \mathsf{Verify}(\mathsf{sk}_V, \mathsf{pk}_V, \mathsf{pk}_S, m, \sigma)$*: Upon input of a verifier's keypair, a sender's public key, a message* m*, and a signature* σ*, outputs the validity of* σ *(a boolean value).*
- $\mathsf{Simulate}_{S\to V}(m) := \mathsf{Simulate}(\mathsf{sk}_V, \mathsf{pk}_V, \mathsf{pk}_S, m)$*: Upon input of a verifier's keypair, a sender's public key, and a message* m*, produces a signature* σ'*.*

We list some relevant properties of DVS schemes, but refer to [JSI96, SKM04] for more details. *Correctness* of a DVS means that for any valid signature $\sigma \leftarrow \mathsf{Sign}_{S\to V}(m)$, $\mathsf{Verify}_{S\to V}(m,\sigma)$ outputs 1 with overwhelming probability. *Unforgeability* of a DVS means that only the signer and the verifier can create a valid signature between them. *Non-transferability* of a DVS ensures that no party can distinguish between valid signatures and their simulations. Lastly, *sender-privacy* of an SDVS guarantees that only the signer and the verifier know the signer's identity and differentiates strong designated verifier schemes from normal designated verifier schemes. Since these last two properties will be used in this work, we give the formal definition in a game-based setting.

Definition 2. *A* DVS *scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Simulate})$ *is* computationally non-transferable *if for any adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\mathsf{NT}}_{\Pi,\mathcal{A}}(\kappa) := \Pr_{b\in\{0,1\}}\left[\mathsf{G}^{\mathsf{NT}}_{\Pi,\mathcal{A}}(\kappa,b)=b\right] - \frac{1}{2} \leq \mathrm{negl}(\kappa),$$

where the game $\mathsf{G}^{\mathsf{NT}}_{\Pi,\mathcal{A}}$ *is defined as follows:*

Algorithm 1: $\mathsf{G}^{\mathsf{NT}}_{\Pi,\mathcal{A}}(\kappa,b)$

$\mathsf{params} \leftarrow \mathsf{Setup}$
$(\mathsf{pk}_S,\mathsf{sk}_S) \leftarrow \mathsf{KeyGen}$
$(\mathsf{pk}_V,\mathsf{sk}_V) \leftarrow \mathsf{KeyGen}$
$(m^*,\mathsf{state}) \leftarrow \mathcal{A}(1,\mathsf{params},\mathsf{pk}_S,\mathsf{sk}_S,\mathsf{pk}_V,\mathsf{sk}_V)$
if $b=0$ **then**
  $\sigma^* = \mathsf{Sign}_{S\to V}(m^*)$
else
  $\sigma^* = \mathsf{Simulate}_{S\to V}(m^*)$
$b' \leftarrow \mathcal{A}(2,\mathsf{state},\sigma^*)$
Output b'

For sender-privacy, we explicitly choose a definition that has been adapted to work in the $n+1$-party setting. For the interested reader we refer to [Wie21] for more information on this choice.

Definition 3 ([Wie21]). *A* DVS *scheme* Π *is* sender-private*, if for any adversary* $\mathcal{A}$ *and any* n*,*

$$\mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}}(\kappa,n) := \Pr_{c\leftarrow\{0,1\}}\left[\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}}(\kappa,n,c)=c\right] - \frac{1}{2} \leq \mathrm{negl}(\kappa),$$

where $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}}$ *is defined as follows:*
Here we use the following oracles:

Algorithm 2: $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi,\mathcal{A}}(\kappa, n, c)$

$\mathsf{params} \leftarrow \mathsf{Setup}$
$(\mathsf{pk}_{P_0}, \mathsf{sk}_{P_0}) \leftarrow \mathsf{KeyGen}; \ldots; (\mathsf{pk}_{P_n}, \mathsf{sk}_{P_n}) \leftarrow \mathsf{KeyGen}$
$(m^*, \mathsf{state}) \leftarrow \mathcal{A}^{\mathcal{O}_{sign}, \mathcal{O}^{(1)}_{veri}, \mathcal{O}_{sim}}(1, \mathsf{params}, \mathsf{pk}_{P_0}, \ldots, \mathsf{pk}_{P_n})$
$\sigma^* = \mathsf{Sign}_{P_c \to P_n}(m^*)$
$c' \leftarrow \mathcal{A}^{\mathcal{O}_{sign}, \mathcal{O}^{(2)}_{veri}, \mathcal{O}_{sim}}(2, \mathsf{state}, \sigma^*)$
Output c'

- $\mathcal{O}_{sign}$*: On input* (m_i, s, v) *returns* $\sigma_i := \mathsf{Sign}_{P_s \to P_v}(m_i)$ *if* $s, v \in [n]$ *and* $\perp$ *otherwise.*
- $\mathcal{O}_{sim}$*: On input* (m_i, s, v) *returns* $\sigma_i := \mathsf{Simulate}_{P_s \to P_v}(m_i)$ *if* $s, v \in [n]$ *and* $\perp$ *otherwise.*
- $\mathcal{O}^{(1)}_{veri}$*: On input* (m_i, σ_i, s, v) *returns* $\mathsf{Verify}_{P_s \to P_v}(m_i, \sigma_i)$ *if* $s, v \in [n]$ *and* $\perp$ *otherwise.*
- $\mathcal{O}^{(2)}_{veri}$*: On input* (m_i, σ_i, s, v) *returns* $\mathsf{Verify}_{P_s \to P_v}(m_i, \sigma_i)$ *if* $s, v \in [n]$ *and* $\sigma_i \neq \sigma^*$*, and* $\perp$ *otherwise.*

In this paper, we will make use of the SDVS scheme proposed in [NJ16], called SUSDVS, which satisfies the above properties when assuming some properties of lattices explained in Section 3.

BB84 In Algorithm 3, we describe the *BB84* protocol, the most well-known QKE variant due to Bennett and Brassard [BB84]. We use the well-established formalism based on error-correcting codes, used by Shor and Preskill [SP00]. Let $C_1[n, k_1]$ and $C_2[n, k_2]$ be two classical linear binary codes encoding k_1 and k_2 bits in n bits such that $\{0\} \subset C_2 \subset C_1 \subset \mathbf{F}_2^n$ where $\mathbf{F}_2^n$ is the binary vector space on n bits. A mapping of vectors $v \in C_1$ to a set of basis states (codewords) for the Calderbank-Shor-Steane (CSS) [CS96, Ste96] code subspace is given by: $v \mapsto (1/\sqrt{|C_2|}) \sum_{w \in C_2} |v + w\rangle$. Due to the irrelevance of phase errors and their decoupling from bit flips in CSS codes, Alice can send $|v\rangle$ along with classical error-correction information $u+v$ where $u, v \in \mathbf{F}_2^n$ and $u \in C_1$, such that Bob can decode to a codeword in C_1 from $(v + \epsilon) - (u + v)$ where ϵ is an error codeword, with the final key being the coset leader of $u + C_2$.

3 Framework

Security Model We use the QKD model from [MSU13] to model the combination of classical and quantum communication, for which we provide a brief overview here. Each *party* in this model has access to both a classical and a quantum Turing machine, connected by a private tape. Furthermore, the classical machine has access to a private randomness tape and both machines can communicate to other parties over public tapes. Two or more parties may execute a *protocol*, which is specified as a series of subroutines. Each subroutine is triggered by an *activation* over one of the tapes. In Appendix A, we present an overview of the exact activations that can be performed.

Algorithm 3: BB84 for an n-bit key with protection against δn bit errors

1 Alice generates two random bit strings $a, b \in \{0,1\}^{(4+\delta)n}$, encodes a_i into $|\psi_i\rangle$ in basis $(+)$ if $b_i = 0$ and in $(\times)$ otherwise, and $\forall i \in [1, |a|]$ sends $|\psi_i\rangle$ to Bob.
2 Bob generates a random bit string $b' \in \{0,1\}^{(4+\delta)n}$ and upon receiving the qubits, measures $|\psi_i\rangle$ in $(+)$ or $(\times)$ according to b'_i to obtain a'_i.
3 Alice announces b and Bob discards a'_i where $b_i \neq b'_i$, ending up with at least $2n$ bits with high probability.
4 Alice picks a set p of $2n$ bits at random from a, and a set q containing n elements of p chosen as check bits at random. Let $v = p \setminus q$.
5 Alice and Bob compare their check bits and abort if the error exceeds a predefined threshold.
6 Alice announces $u + v$, where v is the string of the remaining non-check bits, and u is a random codeword in C_1.
7 Bob subtracts $u + v$ from his code qubits, $v + \epsilon$, and corrects the result, $u + \epsilon$, to a codeword in C_1.
8 Alice and Bob use the coset of $u + C_2$ as their final secret key of length n.

Modeling the Adversary In our work, the main objective of the adversary is to prove the involvement of a party in a key exchange protocol, e.g. to prove that A talked to B. The model, as presented in [MSU13], was mainly used for an eavesdropping adversary, but in our case, we will consider the adversary to always be the initiator (A) or responder (B) in a protocol. The reason for this is that if no adversary $\mathcal{M}$ can prove that A talked to $\mathcal{M}$, then surely $\mathcal{M}$ can also not prove that A talked to B. This argument is also made in [DRGK06], although we provide some alternate views on this in the discussion.

Concretely, this means we model the adversary as a quantum and classical Turing machine, who can perform the QKE protocol with any number of *honest* parties P_i. The adversary can be the initiator or responder in any of these interactions. For any adversary $\mathcal{M}$, we write $\mathsf{View}_{\mathcal{M}}$ to mean the complete contents of $\mathcal{M}$'s memory at the end of execution, including all keys that were established with the other parties.

Security Assumptions The security and deniability of the particular scheme we present rely on several assumptions regarding the quantum safeness of lattice problems, inherited from the SDVS scheme used. In particular, these are the SIS and ISIS problems, which are thought to be quantum secure. In Appendix B we present the exact parameters needed.

4 Deniability

We first provide a natural extension of the classical definition of deniability given in [DRGK06] to the quantum setting, by making both the adversary and the distinguisher a QPT algorithm. In the following definition, the adversary $\mathcal{M}$ is given access to the public keys of an arbitrary amount of honest parties, with whom $\mathcal{M}$ can interact. $\mathcal{M}$ is also given an auxiliary input from the set AUX.

The simulator is given all the same inputs as $\mathcal{M}$, including the same classical randomness, but cannot interact with the honest parties.

Definition 4. *A* QKE *scheme* $(\mathsf{AKG}, \Sigma_I, \Sigma_R)$ *is* deniable *w.r.t.* AUX *if for any* QPT *adversary* $\mathcal{M}$ *there exists a* QPT *simulator* $\mathsf{SIM}_{\mathcal{M}}$ *s.t.*

$$\forall \kappa \in \mathbb{N}, aux \in \mathsf{AUX} : \mathcal{R}eal(\kappa, aux) \approx_q \mathcal{S}im(\kappa, aux),$$

where

$$\mathcal{R}eal(\kappa, aux) = [(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{AKG}(1^\kappa); (aux, \mathbf{pk}, \mathsf{View}_{\mathcal{M}}(\mathbf{pk}, aux))]$$
$$\mathcal{S}im(\kappa, aux) = [(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{AKG}(1^\kappa); (aux, \mathbf{pk}, \mathsf{SIM}_{\mathcal{M}}(\mathbf{pk}, aux))].$$

Definition 5. *Given a public-key signature scheme* $(\mathsf{AKG}, \mathsf{Sign}, \mathsf{Verify})$*, we define the* QKE *scheme* $\mathsf{AuthBB} := (\mathsf{AKG}, \Sigma_I, \Sigma_R)$ *(authenticated BB84), where* Σ_I *and* Σ_R *are as in Algorithm 5 and Algorithm 6 respectively, which can be found in Appendix A. This is the implementation of BB84 as described before, in the above-presented model and using the public-key signature scheme for the classical authentication.*

We restate the definition of deniability, in order to relate deniability to the properties of SDVS, which are presented in a game-based setting.

Definition 6 (Restatement of Definition 4 with $\mathsf{AUX} = \{0\}$**).** *A* QKE *scheme* $\Pi = (\mathsf{AKG}, \Sigma_I, \Sigma_R)$ *is* deniable *if for any* QPT *adversary* $\mathcal{M}$ *there exists a* QPT *simulator* $\mathsf{SIM}_{\mathcal{M}}$ *that does not interact with any party s.t. no* QPT *distinguisher* $\mathcal{F}$ *can achieve non-negligible advantage* $\mathsf{Adv}^{\mathsf{Den}}_{\Pi,\mathcal{F},\mathcal{M},\mathsf{SIM}_{\mathcal{M}}}(\kappa, n)$*, which is the advantage in winning the game* $\mathsf{G}^{\mathsf{Den}}_{\Pi,\mathcal{F},\mathcal{M},\mathsf{SIM}_{\mathcal{M}}}$ *as defined in Algorithm 4.*

Algorithm 4: $\mathsf{G}^{\mathsf{Den}}_{\Pi,\mathcal{F},\mathcal{M},\mathsf{SIM}_{\mathcal{M}}}(\kappa, n, b)$

$(\mathsf{pk}_{P_0}, \mathsf{sk}_{P_0}) \leftarrow \mathsf{AKG}; \ldots; (\mathsf{pk}_{P_{n-1}}, \mathsf{sk}_{P_{n-1}}) \leftarrow \mathsf{AKG}$
Let $\mathbf{pk} = \mathsf{pk}_{P_0} \ldots \mathsf{pk}_{P_{n-1}}$
if $b = 0$ **then**
    $b' \leftarrow \mathcal{F}(\mathsf{View}_{\mathcal{M}}(\mathbf{pk}), \mathbf{pk})$
else
    $b' \leftarrow \mathcal{F}(\mathsf{SIM}_{\mathcal{M}}(\mathbf{pk}), \mathbf{pk})$
Output b'

The advantage of $\mathcal{F}$ *in this game is defined as:*

$$\mathsf{Adv}^{\mathsf{Den}}_{\Pi,\mathcal{F},\mathcal{M},\mathsf{SIM}_{\mathcal{M}}}(\kappa, n) := \Pr_{b \leftarrow \{0,1\}}\left[\mathsf{G}^{\mathsf{Den}}_{\Pi,\mathcal{F},\mathcal{M},\mathsf{SIM}_{\mathcal{M}}}(\kappa, n, b) = b\right] - \frac{1}{2}$$

4.1 Deniable PK-Authenticated BB84

In the following theorem, we provide a concrete scheme that satisfies our version of deniability, however we emphasize that the precise schemes chosen for this (in particular SUSDVS) are simply examples and not critical to the satisfiability of the definition, another possibility could be the scheme presented in [STW12].

Theorem 1. AuthBB *with authentication scheme* Π_{SDVS} *is deniable if* Π_{SDVS} *is an* SDVS *with non-transferability against quantum adversaries.*

Proof. Fix an arbitrary $\mathcal{M}$. This adversary $\mathcal{M}$ can generate many different keypairs to perform protocol sessions with the honest parties, or even use public keys for which they do not know the private key. However, it is not the goal of the adversary to impersonate or trick any of the honest parties, but simply to convince a third-party that they interacted with one of the honest parties. Thus, w.l.o.g. we assume that, for each session, the adversary either uses a keypair $(\mathsf{pk}_{\mathcal{M}}, \mathsf{sk}_{\mathcal{M}})$ for which $\mathcal{M}$ knows the secret key or uses $\mathsf{pk}'_{\mathcal{M}}$ for which $\mathcal{M}$ does not know the private key. $\mathsf{SIM}_{\mathcal{M}}$ simulates $\mathcal{M}$ and all P_i, except:

(*) For each P_i, generate a keypair $(\mathsf{pk}'_{P_i}, \mathsf{sk}'_{P_i})$.
(1) Each call of $\mathsf{Sign}_{P_i \to \mathcal{M}}$ is replaced with $\mathsf{Simulate}_{P_i \to \mathcal{M}}$.
(2) Each call of $\mathsf{Sign}(\mathsf{sk}_{P_i}, \mathsf{pk}_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$ is replaced with $\mathsf{Sign}(\mathsf{sk}'_{P_i}, \mathsf{pk}'_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$.
(3) Each call of Verify is replaced with $\top$.

By definition, each P_i performs only honest executions of the protocol, thus only uses its private key in Sign and Verify. Furthermore, each Sign using sk_{P_i} is replaced with either a Sign using sk'_{P_i} or a $\mathsf{Simulate}$, which does not make use of sk_{P_i}. Each Verify is replaced with a static $\top$, which also does not use sk_{P_i}. This means $\mathsf{SIM}_{\mathcal{M}}$ can run on input $\mathbf{pk}$, i.e. simulate each party P_i without the knowledge of sk_{P_i}.

First we show that change (1) is undetectable by an adversary. Define $P_i^{(1)}$ to be the simulation of P_i after modification (1) and $\mathsf{SIM}_{\mathcal{M}}^{(1)}$ to be the simulation of $\mathcal{M}$ interacting with $P_i^{(1)}$ instead of P_i. Let Π be AuthBB using Π_{SDVS}. For any fixed distinguisher $\mathcal{F}$, let $\mathcal{H}_{start}$ be the $b = 0$ instance of $\mathsf{G}^{\mathsf{Den}}_{\Pi,\mathcal{F},\mathcal{M},\mathsf{SIM}_{\mathcal{M}}^{(1)}}$ and $\mathcal{H}_{end}$ be the $b = 1$ instance. Let $\mathcal{H}_0, \ldots, \mathcal{H}_m$ be a series of hybrids such that $\mathcal{H}_0 = \mathcal{H}_{start}$, $\mathcal{H}_m = \mathcal{H}_{end}$ and each step $\mathcal{H}_k \to \mathcal{H}_{k+1}$ replaces one $\mathsf{Sign}_{P_i \to \mathcal{M}}(x)$ with $\mathsf{Simulate}_{P_i \to \mathcal{M}}(x)$.

Suppose, for some fixed k, there exists a QPT distinguisher $\mathcal{D}$ that can distinguish between $\mathcal{H}_k$ and $\mathcal{H}_{k+1}$, where one $\mathsf{Sign}_{P_i \to \mathcal{M}}(x)$ is replaced in the step $\mathcal{H}_k \to \mathcal{H}_{k+1}$. We use this distinguisher to build an adversary $\mathcal{A}$ that breaks the non-transferability of Π_{SDVS}, as follows:

- $\mathcal{A}$ receives $(1, \mathsf{params}, \mathsf{pk}_S, \mathsf{sk}_S, \mathsf{pk}_V, \mathsf{sk}_V)$.
- $\mathcal{A}$ runs $\mathcal{H}_k$, but replaces $(\mathsf{pk}_{P_i}, \mathsf{sk}_{P_i}, \mathsf{pk}_{\mathcal{M}}, \mathsf{sk}_{\mathcal{M}})$ with $(\mathsf{pk}_S, \mathsf{sk}_S, \mathsf{pk}_V, \mathsf{sk}_V)$ before running $\mathcal{F}$.
- $\mathcal{A}$ stops $\mathcal{H}_k$ before the replaced $\mathsf{Sign}_{P_i \to \mathcal{M}}(x)$ call and outputs (x, state), where state is the state of $\mathcal{A}$ at this point.

– $\mathcal{A}$ receives $(2, \mathsf{state}, \sigma^*)$ and restores from state.
– $\mathcal{A}$ replaces $\mathsf{Sign}_{P_i \to \mathcal{M}}(x)$ with σ^* and continues running $\mathcal{H}_k$.
– $\mathcal{A}$ runs $\mathcal{D}$ and outputs what $\mathcal{D}$ outputs.

Observe that in the $b = 0$ case of $\mathsf{G}^{\mathsf{NT}}_{\Pi_{\mathsf{SDVS}},\mathcal{A}}$, the Sign is replaced by $\mathsf{Sign}_{S \to V}(x)$ and in the $b = 1$ case it is replaced by $\mathsf{Simulate}_{S \to V}(x)$. Furthermore, observe that the insertion of the $(\mathsf{pk}_S, \mathsf{sk}_S, \mathsf{pk}_V, \mathsf{sk}_V)$ keypairs is only a relabeling of the keypairs, but ensures that the $b = 0$ case of $\mathsf{G}^{\mathsf{NT}}_{\Pi_{\mathsf{SDVS}},\mathcal{A}}$ is equal to $\mathcal{H}_k$ and the $b = 1$ case equal to $\mathcal{H}_{k+1}$, thus the distinguishing probability of $\mathcal{D}$ is the same as the winning probability of $\mathcal{A}$ in the $\mathsf{G}^{\mathsf{NT}}_{\Pi_{\mathsf{SDVS}},\mathcal{A}}$ game, which would imply that $\mathsf{Adv}^{\mathsf{NT}}_{\Pi_{\mathsf{SDVS}},\mathcal{A}}$ is non-negligible. Since this is a contradiction, it must be the case that no such $\mathcal{D}$ exists.

For modification (2), the argument is similar. Suppose, in a chain of hybrids similar to the one above, there are two hybrids $\mathcal{H}$ and $\mathcal{H}'$, where $\mathcal{H}'$ is the result of replacing one call of $\mathsf{Sign}(\mathsf{sk}_{P_i}, \mathsf{pk}_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$ with $\mathsf{Sign}(\mathsf{sk}'_{P_i}, \mathsf{pk}'_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$ in $\mathcal{H}$ and some QPT distinguisher $\mathcal{D}$ can distinguish between them. We use this distinguisher to build an adversary $\mathcal{B}$ that breaks the $n+2$-party sender-privacy of Π_{SDVS}, as follows:

– $\mathcal{B}$ receives $(1, \mathsf{params}, \mathsf{pk}_0, \ldots, \mathsf{pk}_{n+1})$.
– $\mathcal{B}$ runs $\mathcal{H}$, but replaces $(\mathsf{pk}_{P_i}, \mathsf{pk}'_{P_i}, \mathsf{pk}_{P_0}, \ldots, \mathsf{pk}_{P_{i-1}}, \mathsf{pk}_{P_{i+1}}, \ldots, \mathsf{pk}_{P_{n-1}}, \mathsf{pk}'_{\mathcal{M}})$ with
$(\mathsf{pk}_0, \ldots, \mathsf{pk}_{n+1})$ before running $\mathcal{F}$. All Sign, $\mathsf{Simulate}$ and Verify calls involving some P_j are performed by oracle calls.
– $\mathcal{B}$ stops $\mathcal{H}$ before the $\mathsf{Sign}(\mathsf{sk}_{P_i}, \mathsf{pk}_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$ call that will be replaced in $\mathcal{H}'$ and outputs (x, state), where state is the state of $\mathcal{A}$ at this point.
– $\mathcal{B}$ receives $(2, \mathsf{state}, \sigma^*)$ and restores from state.
– $\mathcal{B}$ replaces $\mathsf{Sign}(\mathsf{sk}_{P_i}, \mathsf{pk}_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$ with σ^* and continues running $\mathcal{H}$.
– $\mathcal{B}$ runs $\mathcal{D}$ and outputs what $\mathcal{D}$ outputs.

Observe that in the $b = 0$ case of $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi_{\mathsf{SDVS}},\mathcal{B}}$, Sign is replaced by $\mathsf{Sign}(\mathsf{sk}_{P_i}, \mathsf{pk}_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$ and in the $b = 1$ case it is replaced by $\mathsf{Sign}(\mathsf{sk}'_{P_i}, \mathsf{pk}'_{P_i}, \mathsf{pk}'_{\mathcal{M}}, x)$. Furthermore, observe that the replacement of the public keys for all honest parties and pk'_{P_i} is simply a relabeling, since they were all honestly generated. The only public key that is not honestly generated was $\mathsf{pk}'_{\mathcal{M}}$, however since the adversary, by definition, does not know the corresponding private key the replacement is undetectable to the adversary. The replacements of the keys ensures that the $b = 0$ case of $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi_{\mathsf{SDVS}},\mathcal{B}}$ is equal to $\mathcal{H}$ and the $b = 1$ case equal to $\mathcal{H}'$, thus the distinguishing probability of $\mathcal{D}$ is the same as the winning probability of $\mathcal{B}$ in the $\mathsf{G}^{\mathsf{SendPriv}}_{\Pi_{\mathsf{SDVS}},\mathcal{B}}$ game, which would imply that $\mathsf{Adv}^{\mathsf{SendPriv}}_{\Pi_{\mathsf{SDVS}},\mathcal{B}}$ is non-negligible. Since this is a contradiction, it must be the case that no such $\mathcal{D}$ exists.

For modification (3), observe that both the $\mathsf{initiator}$ and the $\mathsf{responder}$ perform their verification after having done all their communication. This means that it is impossible for $\mathcal{M}$ to prove to a third party whether the key exchange was accepted or rejected by P_i. Modification (3) ensures that, for the simulator, even invalid signatures are accepted by the simulated honest parties, but this change cannot alter the behaviour of the simulated adversary as the adversary cannot

detect this change. In fact, for each session the simulated honest party could stop their execution after the last message is sent since the rest of the execution is private and does not influence future sessions.

□

Corollary 1. *Under Assumptions B1 and B2,* SUSDVS *(from [NJ16]) is an* SDVS *with non-transferability against quantum adversaries, thus* AuthBB *using* SUSDVS *is deniable in the standard model.*

4.2 Eavesdropping on Interactions Between Honest Parties

In Theorem 1 we assume that the honest parties only perform QKE sessions with the adversary, arguing that the adversary has no more power as a third-party observer than she has as one of the participants. This assumption was also made in [DRGK06] and we consider it fundamentally sound. However, one can consider what happens if we relax it and give the adversary the ability to force two honest parties to perform a QKE session. The reason that this setting is interesting, is that the simulator is no longer able to create a signature between two honest parties, as doing so requires the private key of either of the honest parties. E.g. if two-party ring signatures were used for authentication, then when Alice and Bob communicate the adversary can prove that at least one of them was present, which would defy deniability.

To solve this problem, one can use the sender-privacy property of an SDVS scheme. The simulator simply generates a keypair for each simulated honest party and uses this to sign any messages, still designating the verifier by their original public key. Since all eavesdropped sessions are between honest parties, the simulator can skip the verification of these signatures. The sender-privacy property ensures that no third party can distinguish between these incorrect signatures and any correct ones the adversary might have collected.

5 Discussion and Future Work

While the work presented here provides a firm basis for deniability in the quantum setting, some obvious open problems remain. Firstly, our protocol delays all authentication until the end. This is done to stop the adversary from intentionally sending an invalid signature to cause an abort, as the simulator would not be able to perform the verification when simulating the honest parties. However, this intentional abort can only be caused by the behaviour of the adversary, which the simulator knows. Thus, intuitively deniability should be achievable without this modification.

Furthermore, in the case of QKE, there is inherent independence between the classical communication and the established key, meaning that the classical part of the transcript contains no information about the established key [MSU13]. This leads us to conjecture that using any deniable public-key authentication might be enough to create a deniable QKE protocol.

Finally, we limit ourselves to the case where $\mathsf{AUX} = \{0\}$ for simplicity, however we conjecture that this restriction is not necessary and that the provided proof extends to any AUX.

A Appendix: The [MSU13] Model

In this appendix we elaborate on the [MSU13] model. The classical Turing machine can receive the following activations:

- $\mathsf{SendC}(\Psi, \mathsf{msg})$: The Turing machine resumes the *session* with identifier Ψ using msg as input. Ψ may also be a vector of session identifiers, where it is clear from context which one belongs to the receiving party and which to other parties.
- $\mathsf{SendC}(\mathsf{params}, \mathsf{pid})$: When SendC is received without a session identifier it indicates the start of a new protocol execution with public parameters params.
- $\mathsf{Q2C}(\mathsf{msg})$: This activation indicates a classical output of the quantum Turing machine and activates the classical Turing machine with the most recent session.

The quantum Turing machine has the following activations:

- $\mathsf{SendQ}(\rho)$: The quantum Turing machine activates with as input the state ρ.
- $\mathsf{C2Q}(\mathsf{msg})$: The quantum Turing machine is activated by the classical Turing machine with message msg.

We use Ψ to denote *ephemeral* variables, which are variables that are bound to a session. After each activation, the Turing machines may send activations over their respective public channel and the private channel between them. At the end of a session, the classical Turing machine of both parties outputs four values:

- sk, the shared secret key established during this session, or $\perp$ if execution failed.
- pid, the identifier of the other party involved in this session.
- A vector $\boldsymbol{v} = (\boldsymbol{v_0}, \dots)$, where each $\boldsymbol{v_i}$ is a vector of labels of values.
- A vector $\boldsymbol{u} = (\boldsymbol{u_0}, \dots)$, where each $\boldsymbol{u_i}$ is a vector of labels of values.

A protocol is *correct* if, when all messages are delivered without changes or reordering, both parties output the same key sk and the same vector $\boldsymbol{v}$. Each classical value Ψ_d has a label $\ell(\Psi_d)$ and an adversary can partner a value by issuing $\mathsf{Partner}(\ell(\Psi_d))$ to learn the value corresponding to the label. An adversary can also partner a session Ψ, learning the value sk if it has been output. Note that if an adversary learns a value without partnering (through public communication, for example), this value remains *unpartnered*. A session Ψ is *fresh* as long as every $\boldsymbol{v_i}$ contains at least (the label of) one value that the adversary has not partnered and the adversary has not partnered Ψ or any session Ψ' with the same $\boldsymbol{v}$ and sk and, *at the time of output*, there is least one value in each $\boldsymbol{u_i}$ with which the adversary has not partnered. This signifies the main difference between $\boldsymbol{v}$ and $\boldsymbol{u}$: values in $\boldsymbol{u}$ pose no security risk if revealed after the key has been established, but values in $\boldsymbol{v}$ do.

A.1 BB84 in This Model

In Algorithms 5 and 6 we present, respectively, the initiator and responder roles in the [BB84] QKD protocol, following the [MSU13] model.

Algorithm 5: Σ_I

Upon Activation: $\mathsf{SendC}(\mathsf{start}, \mathsf{initiator}, R)$

1.1 Create a new session Ψ^I with responder identifier R

1.2 Read n_1 random bits Ψ^I_{dIR}

1.3 Read n_1 random bits Ψ^I_{bI}

1.4 Send activation $\mathsf{C2Q}(\Psi^I_{dIR}, \Psi^I_{bI}, R)$

1.5 Send activation $\mathsf{SendC}(\Psi^I, \mathsf{start}, \mathsf{responder}, I)$ to R

Upon Activation: $\mathsf{C2Q}(\Psi^I_{dIR}, \Psi^I_{bI}, R)$

2.1 Prepare ρ to be the bitwise encoding of Ψ^I_{dIR} in the $(+)$ or $(\times)$ basis if the corresponding bit of Ψ^I_{bI} is 0 or 1 respectively

2.2 Send activation $\mathsf{SendQ}(\rho)$ to R

Upon Activation: $\mathsf{SendC}(\Psi^I, \Psi^R, \Psi^R_{bR})$

3.1 Discard all bit positions from Ψ^I_{dIR} for which Ψ^I_{bI} is not equal to Ψ^R_{bR}; Let n_2 denote the amount of bits left in Ψ^I_{dIR}

3.2 Read n_2 random bits Ψ^I_{indIR}; Let Ψ^I_{chkIR} be the substring of Ψ^I_{dIR} for which the bits of Ψ^I_{indIR} are 1 and Ψ^I_{kIR} the substring for which they are 0; Let n_3 be the length of Ψ^I_{kIR}

3.3 Send activation $\mathsf{SendC}(\Psi^I, \Psi^R, \Psi^I_{bI}, \Psi^I_{indIR}, \Psi^I_{chkIR})$ to R

Upon Activation: $\mathsf{SendC}(\Psi^I, \Psi^R, \varepsilon, \sigma_R)$

4.1 Read random bits Ψ^I_F to construct a 2-universal hash function F and compute $F' \leftarrow F(\Psi^I_{kIR})$

4.2 Read random bits $\Psi^I_{P,G}$ to construct a 2-universal hash function G and a random permutation P and compute $\Psi^I_{skIR} \leftarrow G(P(\Psi^I_{kIR}))$

4.3 Compute $\sigma_I \leftarrow \mathsf{Sign}(\Psi_I, \Psi_R, \Psi^I_{bI}, \Psi^I_{indIR}, \Psi^I_{chkIR}, F, F', P, G, I)$

4.4 Send activation $\mathsf{SendC}(\Psi^I, \Psi^R, F, F', P, G, \sigma_I)$ to R

4.5 Abort if $\mathsf{Verify}(\sigma_R, (\Psi^I, \Psi^R, \Psi^R_{bR}, \varepsilon, R))$ fails

4.6 Output $(\mathsf{sk} = \Psi^I_{skIR}, pid = R, \boldsymbol{v} = (\ell(\Psi^I_{dIR}), \ell(\Psi^R_{dIR}), \ell(\Psi^I_{bI}), \ell(\Psi^R_{bR}), \ell(\Psi^I_F), \ell(\Psi^I_{P,G})), \boldsymbol{u} = (\mathsf{sk}_I))$

B Appendix: Lattice Hardness Problems Needed for [NJ16]

In this appendix, we briefly present the following assumptions, which are conjectured to hold in the presence of quantum computers, but refer to [NJ16] for their precise statements. We use the following (simplified) parameters:

– $|\mathsf{msg}|$ is the length of the message being signed,

Algorithm 6: Σ_R

Upon Activation: $\mathsf{SendC}(\Psi^I, \mathsf{start}, \mathsf{responder}, R)$

1.1 Create a new session Ψ^R with initiator identifier I

1.2 Read n_1 random bits Ψ^R_{bR}

1.3 Send activation $\mathsf{C2Q}(\Psi^R_{bR})$

Upon Activation: $\mathsf{C2Q}(\Psi^R_{bR})$ combined with $\mathsf{SendQ}(\rho)$

2.1 Set Ψ^R_{dIR} to be the qubit-wise measurement of ρ in the (+) or (×) basis if the corresponding bit of Ψ^R_{bR} is 0 or 1 respectively

2.2 Send activation $\mathsf{Q2C}(\Psi^R_{dIR})$

Upon Activation: $\mathsf{Q2C}(\Psi^R_{dIR})$

3.1 Send activation $\mathsf{SendC}(\Psi^I, \Psi^R, \Psi^R_{bR})$ to I

Upon Activation: $\mathsf{SendC}(\Psi^I, \Psi^R, \Psi^I_{bI}, \Psi^I_{indIR}, \Psi^I_{chkIR})$

4.1 Discard all bit positions from Ψ^R_{dIR} for which Ψ^I_{bI} is not equal to Ψ^R_{bR}

4.2 Let Ψ^R_{chkIR} be the substring of Ψ^R_{dIR} for which the bits of Ψ^I_{indIR} are 1 and Ψ^R_{kIR} the substring for which they are 0

4.3 Let ε be the proportion of bits of Ψ^I_{chkIR} that do not match Ψ^R_{chkIR}; abort if $\varepsilon > 0.061$

4.4 Compute $\sigma_R \leftarrow \mathsf{Sign}(\Psi^I, \Psi^R, \Psi^R_{bR}, \varepsilon, R)$

4.5 Send activation $\mathsf{SendC}(\Psi^I, \Psi^R, \varepsilon, \sigma_R)$ to I

Upon Activation: $\mathsf{SendC}(\Psi^I, \Psi^R, F, F', P, G, \sigma_I)$

5.1 Abort if $\mathsf{Verify}(\sigma_I, (\Psi_I, \Psi_R, \Psi^I_{bI}, \Psi^I_{indIR}, \Psi^I_{chkIR}, F, F', P, G, I))$ fails

5.2 Use F and F' to correct Ψ^R_{kIR} to $\Psi^R_{kIR'}$

5.3 Compute $\Psi^R_{skIR} \leftarrow G(P(\Psi^R_{kIR'}))$

5.4 Output $(\mathsf{sk} = \Psi^R_{skIR}, pid = I, \boldsymbol{v} = (\ell(\Psi^I_{dIR}), \ell(\Psi^R_{dIR}), \ell(\Psi^I_{bI}), \ell(\Psi^R_{bR}), \ell(\Psi^I_F), \ell(\Psi^I_{P,G})), \boldsymbol{u} = (\mathsf{sk}_R))$

- κ is the security parameter,
- $h = O(\log \kappa)$
- $m = O(\kappa h)$,
- $q = \mathsf{poly}(\kappa)$ is a sufficiently large number,
- $l \leq (p-1)\kappa$, where p is the smallest prime dividing q, and
- $s = O(\sqrt{\kappa l h}) \cdot \omega(\sqrt{\log \kappa})^2$ a sufficiently large parameter.

Assumption B1 *The $\mathsf{SIS}_{q,m,\beta}$ problem is hard for sufficiently large $q = \sqrt{(|\mathsf{msg}| + 4ms^2)\kappa} + \omega(\sqrt{\log \kappa})$ and $\beta = \sqrt{|\mathsf{msg}| + 4ms^2}$. The $\mathsf{SIS}_{q,m,\beta}$ problem and $\mathsf{ISIS}_{q,m,\beta}$ problem are hard for sufficiently large $q = O(l^{3/2}\kappa^3 \log^{5/2} \kappa) \cdot \omega(\sqrt{\log \kappa})^6$, $m = O(\kappa \log q)$, and $\beta = s\sqrt{2m}O(l\kappa^{3/2}k^{3/2}) \cdot \omega(\sqrt{\log \kappa})^3$.*

Furthermore, we have the following assumption on the hardness of distinguishing lattices, where q is prime. Here $\mathcal{D}_{\Lambda^\perp_{\boldsymbol{w}}(A),s}$ denotes the distribution of sampling from $\{\boldsymbol{z} \in \mathbb{Z}^\kappa \mid A\boldsymbol{z} = \boldsymbol{w} (\mathrm{mod}\ q)\}$ according to a Gaussian distribution.

Assumption B2 (Assumption 2.1 in [NJ16]) *Let $m_1, m_2 = O(\kappa \log q)$, A, R uniform random matrices from $\mathbb{Z}_q^{\kappa \times m_1}$, $C_0, \ldots, C_l$ uniform random matrices from $\mathbb{Z}_q^{\kappa \times m_2}$, $\boldsymbol{w}$ a fixed vector from $\mathbb{Z}_q^\kappa$, $\mu \in \{0,1\}^l$ a secret bitstring, and $C_\mu = C_0 + \sum_{j=1}^l \mu_j C_j$, then it is hard to distinguish between $\mathcal{D}_{\Lambda^\perp_{\boldsymbol{w}}(A|C_\mu),s}$ and $\mathcal{D}_{\Lambda^\perp_{\boldsymbol{w}}(R|C_\mu),s}$ without any information on μ.*

References

[Ata+18] Arash Atashpendar et al. "Revisiting Deniability in Quantum Key Exchange". In: *Secure IT Systems, 23rd Nordic Conference, NordSec 2018*. Cham: Springer International Publishing, 2018, pp. 104–120. ISBN: 978-3-030-03638-6.

[Ata19] Arash Atashpendar. "From Information Theory Puzzles in Deletion Channels to Deniability in Quantum Cryptography". PhD thesis. University of Luxembourg, Luxembourg, 2019. URL: `https://arxiv.org/pdf/2003.11663.pdf`.

[BB84] Charles H Bennett and Gilles Brassard. "Quantum cryptography: public key distribution and coin tossing Int". In: *Conf. on Computers, Systems and Signal Processing (India, Dec. 1984)*. 1984, pp. 175–9.

[Bea02] Donald Beaver. "On Deniability in Quantum Key Exchange". In: *Advances in Cryptology — EUROCRYPT 2002*. Springer Berlin Heidelberg, 2002, pp. 352–367. ISBN: 978-3-540-46035-0.

[Can+97] Ran Canetti et al. "Deniable Encryption". In: *Advances in Cryptology — CRYPTO '97*. Springer Berlin Heidelberg, 1997, pp. 90–104. ISBN: 978-3-540-69528-8.

[CG96] R. Canetti and R. Gennaro. "Incoercible multiparty computation". In: *Proceedings of 37th Conference on Foundations of Computer Science*. Oct. 1996, pp. 504–513. DOI: `10.1109/SFCS.1996.548509`.

[Che+21] Yu-Ao Chen et al. "An integrated space-to-ground quantum communication network over 4,600 kilometres". In: *Nature* 589.7841 (2021), pp. 214–219.

[CS96] A Robert Calderbank and Peter W Shor. "Good quantum error-correcting codes exist". In: *Physical Review A* 54.2 (1996), p. 1098.

[DNS04] Cynthia Dwork, Moni Naor, and Amit Sahai. "Concurrent Zero-knowledge". In: *J. ACM* 51.6 (Nov. 2004), pp. 851–898. ISSN: 0004-5411. DOI: `10.1145/1039488.1039489`.

[DRGK06] Mario Di Raimondo, Rosario Gennaro, and Hugo Krawczyk. "Deniable Authentication and Key Exchange". In: *Proceedings of the 13th ACM Conference on Computer and Communications Security*. CCS '06. Alexandria, Virginia, USA: ACM, 2006, pp. 400–409. ISBN: 1-59593-518-5. DOI: `10.1145/1180405.1180454`.

[Eur] *European Quantum Communication Infrastructure (EuroQCI) — Shaping Europes digital future*. `https://digital-strategy.ec.europa.eu/en/policies/european-quantum-communication-infrastructure-euroqci`. (Accessed on 09/07/2021).

[IM11] Lawrence M. Ioannou and Michele Mosca. "A New Spin on Quantum Cryptography: Avoiding Trapdoors and Embracing Public Keys". In: *Post-Quantum Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 255–274. ISBN: 978-3-642-25405-5.

[JSI96] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. "Designated Verifier Proofs and Their Applications". In: *Advances in*

Cryptology — EUROCRYPT '96. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 143–154. ISBN: 978-3-540-68339-1.

[MSU13] Michele Mosca, Douglas Stebila, and Berkant Ustaoğlu. "Quantum Key Distribution in the Classical Authenticated Key Exchange Framework". In: *Post-Quantum Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 136–154. ISBN: 978-3-642-38616-9.

[NJ16] Geontae Noh and Ik Rae Jeong. "Strong designated verifier signature scheme from lattices in the standard model". In: *Security and Communication Networks* 9.18 (2016), pp. 6202–6214.

[SKM04] Shahrokh Saeednia, Steve Kremer, and Olivier Markowitch. "An Efficient Strong Designated Verifier Signature Scheme". In: *Information Security and Cryptology - ICISC 2003*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 40–54. ISBN: 978-3-540-24691-6.

[SN17] National Institute of Standards and Technology (NIST). *Post-Quantum Cryptography Standardization*. `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization`. [Online; accessed 22-July-2019]. 2017.

[SP00] Peter W Shor and John Preskill. "Simple proof of security of the BB84 quantum key distribution protocol". In: *Physical review letters* 85.2 (2000), p. 441.

[Ste96] Andrew Steane. "Multiple-particle interference and quantum error correction". In: *Proc. R. Soc. Lond. A* 452.1954 (1996), pp. 2551–2577.

[STW12] X. Sun, H. Tian, and Y. Wang. "Toward Quantum-Resistant Strong Designated Verifier Signature from Isogenies". In: *2012 Fourth International Conference on Intelligent Networking and Collaborative Systems*. Sept. 2012, pp. 292–296. DOI: 10.1109/iNCoS.2012.70.

[Wat18] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018. URL: `https://cs.uwaterloo.ca/~watrous/TQI/TQI.pdf`.

[Wie21] Jeroen van Wier. "On SDVS Sender Privacy In The Multi-Party Setting". In: *CoRR* abs/2107.06119 (2021). arXiv: 2107.06119. URL: `https://arxiv.org/abs/2107.06119`.

[Sho94] Peter W Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. Nov. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.