Quantum Key Distribution in the Classical Authenticated Key Exchange Framework

Michele Mosca[1,2], Douglas Stebila[3], and Berkant Ustaoğlu[4]

[1] Institute for Quantum Computing and Dept. of Combinatorics & Optimization
University of Waterloo, Waterloo, Ontario, Canada
[2] Perimeter Institute for Theoretical Physics, Waterloo, Ontario, Canada
mmosca@uwaterloo.ca
[3] Information Security Discipline, Queensland University of Technology, Brisbane, Queensland, Australia
stebila@qut.edu.au
[4] Department of Mathematics, Izmir Institute of Technology, Urla, Izmir, Turkey
bustaoglu@uwaterloo.ca

Abstract. Key establishment is a crucial primitive for building secure channels in a multi-party setting. Without quantum mechanics, key establishment can only be done under the assumption that some computational problem is hard. Since digital communication can be easily eavesdropped and recorded, it is important to consider the secrecy of information anticipating future algorithmic and computational discoveries which could break the secrecy of past keys, violating the secrecy of the confidential channel.

Quantum key distribution (QKD) can be used generate secret keys that are secure against any future algorithmic or computational improvements. QKD protocols still require authentication of classical communication, although existing security proofs of QKD typically assume idealized authentication. It is generally considered folklore that QKD when used with computationally secure authentication is still secure against an unbounded adversary, provided the adversary did not break the authentication during the run of the protocol.

We describe a security model for quantum key distribution extending classical authenticated key exchange (AKE) security models. Using our model, we characterize the long-term security of the BB84 QKD protocol with computationally secure authentication against an eventually unbounded adversary. By basing our model on traditional AKE models, we can more readily compare the relative merits of various forms of QKD and existing classical AKE protocols. This comparison illustrates in which types of adversarial environments different quantum and classical key agreement protocols can be secure.

Keywords: quantum key distribution, authenticated key exchange, cryptographic protocols, security models

1 Introduction

Quantum key distribution (QKD) promises new security properties compared to cryptography based on computational assumptions: two parties can establish a key using a pair of quantum and classical channels, secure against any adversary who is limited solely by the laws of quantum mechanics. Most information-theoretically secure classical[5] cryptographic tasks have limited practicality, so many schemes' security rely on computational assumptions, the most widely used of which—factoring, discrete logarithms—could be efficiently solved by a large-scale quantum computer. As a result, QKD could be an important primitive for cryptography secure against advances in computing technology, provided quantum mechanics remains an accurate description of the laws of nature.

The classical cryptographic literature has extensively studied *authenticated key exchange* (AKE) since the founding of public key cryptography in 1976. After a period of ad hoc security analysis, protocols are now generally analyzed in a security model where an active attacker controls communication and can possibly compromise certain private information; proofs usually consist of probabilistic reductions to computationally hard problems. The seminal work in this area by Bellare and Rogaway [1] was followed by the more modern CK01 [2] and eCK [3] models; an alternative approach to this family of security models is given by Canetti's *universal composability (UC) framework* [4]. Typically in AKE protocols, calculating a secret key is relatively easy, but authentication—ensuring that the key is shared only with only the intended party—requires greater care.

There are many types of QKD protocols, but for our purposes we will divide them into 3 classes: prepare-send-measure protocols, measure-only protocols, and prepare-send-only protocols. The first QKD protocol, now called BB84 [5], is an example of a prepare-send-measure protocol in which Alice randomly prepares one of several quantum states, sends it to Bob, and Bob randomly measures in one of several settings. Ekert [6] proposed an entanglement-based protocol, which is an example of a measure-only protocol: Alice and Bob only randomly measure in one of several settings; the state itself can be prepared by Eve entirely untrusted. Biham et al. [7] proposed a prepare-send-only protocol, in which Alice and Bob each randomly prepare one of several quantum states and send them to Eve, who measures and sends back a classical result. Different versions can be appealing due to ease of implementation, resistance to side-channel attacks on preparing or measuring, or device independence.

Research on QKD security has largely proceeded independent of the aforementioned classical AKE security models. Various proofs of QKD have been given in a stand-alone 2-party setting [8,9,10,11,12,13,14]. This contrasts with the aforementioned security models used in classical AKE protocols, which consider the multi-party, multi-session setting, and consider various types of information leakage or compromise. Existing QKD proofs typically take place under the assumption that classical communication happens over on authentic public

[5] We use the adjective "classical" to mean "non-quantum", so "classical cryptography" means "non-quantum cryptography", not "historical cryptography".

channel. It is generally considered folklore [15,16,17,18] that if QKD was performed using a computationally secure message authentication scheme (such as public key digital signatures), then messages encrypted under the keys output by QKD would be secure provided that the adversary could not break the authentication scheme *before or during* the QKD protocol. This result has only been justified formally in this paper and in our concurrent work by Unruh in the universal composability setting [19].

Contributions. Our goal is to describe the security of quantum key distribution in a security model similar to existing classical authenticated key exchange protocols and compare the relative security properties of various QKD and classical AKE protocols. Our model is explicitly a multi-party model, includes authentication, and allows for either computationally secure or information theoretically secure authentication. We aim to capture two properties: (1) QKD is *immediately secure* against an active adversary who is restricted such that he is unable to break the authentication scheme, and (2) QKD is *long-term secure*, meaning that, if it is secure against an active adversary who is restricted during the run of the protocol to be unable to break the authentication scheme, then it remains secure even when the (classical and quantum) data obtained by the active bounded adversary are later given to an unbounded quantum adversary.

Security model for classical-quantum AKE protocols. We first introduce in Section 2 a multi-party model for analyzing the security of QKD protocols. In our model, which adopts the formalism of Goldberg et al.'s framework for AKE [20], parties consist of a pair of classical and quantum Turing machines, each of which is capable of sending and receiving messages. The adversary controls all communications between parties, but is restricted in its ability to affect communication between a party's classical and quantum devices. The adversary also has the ability to compromise various values used by parties before, during, or after the run of the protocol. As is typical, the adversary's goal is to distinguish the session key of a completed session from a random string of the same length. A novelty of our approach is a new technique for defining matching sessions.

Having defined the adversarial model, we then introduce our two security definitions, *immediate security* against an active, potentially bounded adversary, and *long-term security*, against an adversary who during the run of the protocol may be bounded, but after the protocol completes is unbounded (except by the laws of quantum mechanics). Our model is generic enough to allow the bound on the adversary to be computational—assuming that a particular computational problem is hard—or run-time or memory-bounded [21]. We adapt the long-term security notion of Müller-Quade and Unruh [22] from the classical universal composability framework to our classical-quantum model.

Security of BB84. We then proceed in Section 3 to show that the BB84 protocol, when used with a computationally secure classical authentication scheme such as a digital signature, is secure in this model. For the quantum aspects of the proof, we rely on existing proof techniques. This is next extended to provide a proof of the folklore theorem that QKD, when used with computationally secure authentication in a multi-party setting, is information theoretically se-

cure, provided the adversary did not break the authentication during the run of the protocol. Our argument explicitly identifies which secret information leakage does not affect security either before or after the run of the protocol.

Comparison of quantum and classical AKE protocols. Finally, we use our generic security model to compare in Section 4 the security properties of classical key exchange protocols and examples from each of the three classes of QKD protocols (prepare-send-measure, measure-only, prepare-send-only). This comparison is facilitated by our phrasing of QKD in a security model more closely related to traditional AKE security models, which we can then use to compare the relative powers afforded to the adversary under those models. In particular, our model allows us to compare how different protocols react when the randomness used in the protocol is revealed—or if it is later discovered that bad randomness was used. For example, some classical AKE protocols such as `UP` [23] are secure even if the randomness used for either a party's long-term secret key or ephemeral secret key is revealed *before* the run of the protocol, but the same is not true for the randomness used to pick basis choices in BB84. And the EPR protocol of Ekert is secure even if all of the randomness used by the parties is leaked after the protocol completes, unlike BB84 where data bit choices must remain secret. Since obtaining high quality randomness can be very challenging in practice—requiring either a separate, tested quantum source, or relying on a pseudorandom number generator seeded from a high quality source of entropy—it may be desirable to select a protocol based on the quality of randomness available, and our framework provides a method for comparing protocols along these lines.

Comparison with other frameworks. Our approach to defining security differs from existing work in several essential ways. Stand-alone QKD security definitions do not consider the security in a *multi-party setting*, and also tend to ignore entirely the question of *explicit authentication*, instead assuming an authentic classical channel. It is widely recognized that the authentication can be secure against an unbounded adversary if all classical communication is protected by information-theoretically secure message authentication codes, such as the Wegman-Carter 2-universal hash function [24,25]. However, as mentioned above, the classical AKE experience suggests that it is the authentication part of the overall security definition that is often violated; more so when there is *information leakage* to adversary. With a few exceptions (e.g., [13]), stand-alone definitions also exclude the possibility of the adversary learning private information. The universal composability definition of QKD security of Ben-Or et al. [26] (which is an adaptation of Canetti's UC framework [4] to the quantum setting), notably referenced by Renner in his thesis [14], also brushes aside the possibility of any information being leaked to the adversary and focuses solely on information-theoretic authentication. Other frameworks for composability of quantum protocols have been given [27,28,29,30,31] and applied to other types of cryptographic protocols, but not QKD. Our model, then, is the first to define QKD security in the multi-party setting, with explicit consideration of authentication, allowing leakage of information the adversary. Moreover, it defines both

short-term and long-term security; last but not least our definitions paves way for formally analyzing and comparing both classical and quantum AKE protocols within the same framework. In work concurrent with our, Unruh [19] analyzes the long-term security of QKD in the UC framework.

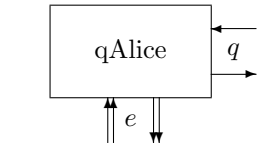
2 QKD model

Our model begins as an enhancement to the eCK model [3], following the notation of Goldberg et al. [20]. In our model, each party has access to a quantum device. The quantum device may be viewed as limited based on for example current hardware limitations. As usual we consider interactive protocols within a multi-party multi-session setting, where communication is controlled by the adversary. The adversary controls the quantum communication channel between parties, subject to the laws of quantum physics. We also describe how, if at all, the adversary may gain access to secrets used by the parties. We then define secrecy against bounded adversaries and long-term security against unbounded adversaries: the long-term security definition is achieved by having the active bounded short-term adversary output a classical and quantum transcript upon which the unbounded quantum adversary may operate.

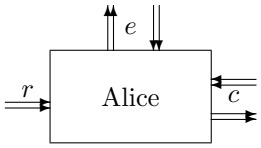
We next formally describe the model. We use k to denote a security parameter. Our description uses qubits but can be generalized to arbitrary-dimension quantum systems.

2.1 Parties and protocols

A *party* (see also [32, Def. 1.1, bullet 2]) is an interactive classical Turing machine with access to a quantum Turing machine. We refer to this pair jointly as the party.



(a) Quantum Turing machine



(b) Classical Turing machine

Fig. 1. A party's classical and quantum Turing machines

The classical machine can activate the quantum device via a special activation request or receive (via designated activation routines) measurement outcomes from the quantum device. The communication is delivered over a two way classical communication tape (the e-channel in Figure 1(b)). The classical Turing machine has also access to a sequence of random bits – the r-tape in Figure 1(b) – and a separate c-tape over which the party can receive and send other activation requests and messages as specified by designated routines. Similarly, the quantum Turing device can be activated by the classical Turing machine and can receive and send qubits over a designated quantum channel q as in Figure 1(a).

Each party can have associated authenticated public strings (e.g., public keys or identifiers), which are assumed to be distributed over an authenticated channel

to other parties. Furthermore, pairs of parties may possess shared secrets that were distributed confidentially a priori.

A *protocol* is a collection of interactive classical and quantum subroutines that produce a shared secret key between two (or more parties) or output an indicator of an error. The interactions may use messages received on either the classical or quantum channels. The final output of the protocol is made via the classical Turing machine.

A *session* is an execution of the protocol. Sessions are initiated via a special incoming request and upon initiation each one is identified with a unique[6] *session identifier* Ψ chosen by the party at which the session is executed (in which case we say the party *owns* the session). A session that has been initiated but is not yet completed is called *active*. Since sessions are interactive procedures a party may own more than one active session at a given point of time. Each active session has a separate *session state* that stores session-specific classical data.[7]

Upon receiving and sending all protocol messages and performing the required measurements and computations specified by the protocol, the session *completes* by having the classical Turing machine output either an error symbol $\perp$ or a tuple $(sk, pid, \mathbf{v}, \mathbf{u})$. The tuple consists of:

- sk: a session key;
- pid: a party identifier;
- $\mathbf{v}$: a vector $(\mathbf{v}_0, \mathbf{v}_1, \dots)$ where each $\mathbf{v}_i$ is a vector of public values or labels; (For example, $\mathbf{v}_1$ may consist of the public values contributed by party P_1. Including $\mathbf{v}$ in the session output binds the session with the various values used by the parties to compute the session key.)
- $\mathbf{u}$: a vector $(\mathbf{u}_0, \mathbf{u}_1, \dots)$ where each $\mathbf{u}_i$ is vector of a public values or labels; $\mathbf{u}$ is called the *authentication* vector and indicates what the session owner uses to identify its peer pid.

The vectors $\mathbf{v}$ and $\mathbf{u}$ will play an important role in defining freshness.

Definition 1 (Correctness). *A key exchange protocol π is* correct *if, when all protocol messages are relayed faithfully, without changes to content or ordering, the peer parties output the same session key k and the same vector $\mathbf{v}$.*

Memory. A party may hold in its memory several *value pairs* of the form (x, X), generated by some algorithm specified by the protocol, where x is a private value and X is a public value or label. The pair may be a *public key pair*, such as private key x and public key X, or a *labelled private value*, such as a private value x and a unique public label $X = \ell(x)$.

[6] With this definition uniqueness is guaranteed only within a party; globally uniqueness can be guaranteed by requiring the session identifier is the concatenation of the unique party identifier and the party's own session identifier.

[7] While quantum protocols in general may make use of quantum memory for storing quantum states during a session, the current QKD protocols we consider in this paper, such as BB84 or EPR, do not, so we omit this from our model.

There are two classifications of value pairs: *ephemeral* value pairs, which are associated with a particular session Ψ, and *static* value pairs, which can be used across multiple sessions. The party may also have value pairs that have been generated but not yet used. If necessary, different types of key pairs may be permitted, for example, if a protocol uses one type of key pair for digital signatures and another type of key pair for public-key encryption. The protocol specifies an algorithm for generating new pairs.

Classical Turing machine communication. As described above each classical Turing machine has two incoming-outgoing classical communication channels, denoted by e and c in Figure 1(b), over which the classical Turing machine receives activations and submits responses. The responses themselves can be activation requests. Furthermore the classical Turing machine has an input of classical random bits which can be read at will by the Turing machine, denoted by r in Figure 1(b). The following activations of the classical Turing machine are allowed:

- $\mathsf{SendC}(params, pid)$: This activation is received via channel c and directs the party to begin a new key exchange session. A new session is initiated and assigned a unique session identifier Ψ based on protocol-specific public parameters $params$ and an identifier pid of the party with whom to establish the session. The response to this query includes the session identifier Ψ and any protocol-specific outgoing classical message msg' that are sent via the outgoing channel c. If required by the protocol, the Turing machine can send an activation request $\mathsf{C2Q}(m)$ over the e outgoing channel, which may in turn cause that quantum Turing machine to write an output to its q channel as well, or to prepare its measurement device to receive quantum messages.
- $\mathsf{SendC}(\Psi, msg)$: This query models the delivery of classical messages over c-channel. The party's classical Turing machine is activated with session Ψ and classical message msg. It returns any outgoing classical message msg' over the c-channel. If required by the protocol, the Turing machine can send an activation request $\mathsf{C2Q}(m)$ over the e outgoing channel, which may in turn cause that quantum Turing machine to write an output to its q channel as well, or to prepare its measurement device to receive quantum messages.
- $\mathsf{Q2C}(m)$: Upon activation with this query the classical Turing machine activates its most recent session with input m. This query may cause the classical Turing machine to output to its c channel, or send another activation over the e channel.

A protocol may request that the classical Turing machine acts probabilistically, in which case it reads random bits from the r-channel.

Quantum Turing machine communication. Each party's quantum Turing machine has a two-way quantum communication channel, denoted by q in Figure 1(a), over which the machine receives and submits quantum information. The responses themselves can be activation requests. Furthermore the quantum Turing machine has a two-way classical control channel (denoted by e in Figure 1(a)) with which it communicates with the classical Turing machine.

The following activations of the quantum Turing machine are allowed:

- $\mathsf{SendQ}(\rho)$: This query activates the quantum Turing machine with quantum message ρ; it returns any outgoing quantum message ρ' over the q-channel. If required by the protocol, the quantum Turing machine can send an activation request $\mathsf{C2Q}(m)$ over the e outgoing channel (for example, to report any measurement results obtained from measuring ρ), which may in turn cause that classical Turing machine to write an output to its c channel as well.
- $\mathsf{C2Q}(m)$: This query activates the quantum Turing machine with classical control message m, for example to prepare the quantum circuit for execution due to an anticipated SendQ activation. The activation may cause a quantum state to be output over the outgoing quantum channel q as well as a classical message to be returned over classical control channel e.

2.2 Adversarial model

The *adversary* is, similar to a party, a pair of interactive classical and quantum Turing machines. The adversary's classical Turing machine runs in time at most $t_c(k)$ and has access to a quantum Turing machine with runtime bounded by $t_q(k)$ and memory bounded by $m_q(k)$ qubits; bounds may be unlimited. The adversary takes as its input all public information and may interact with the (honest) parties. Furthermore the adversary can establish corrupted (dishonest) parties which it fully controls. Honest parties cannot distinguish between honest and dishonest parties.

Communication over the parties' classical c-channels is controlled by the adversary. On the classical channels, the adversary can read, copy, reorder, insert, delay, modify, drop or forward messages at will. The sending and receiving parties have no intrinsic mechanism to detect which actions, if any, the adversary performed on the classical messages.

Communication over the parties' quantum q channels is also controlled by the adversary. The adversary's operations on the quantum channels are bound by the laws of quantum mechanics: the delivery of quantum messages can be delayed, modified in order, forwarded, or dropped; the adversary can create new quantum states and perform joint quantum operations on quantum messages received from the parties as well as on the adversary's state. However, due to the laws of quantum mechanics, the adversary cannot necessarily obtain full information about quantum messages from the parties; for example, measurements may irrevocably disturb the state of messages transmitted by the parties, and the adversary may be unable to precisely copy a message due to the no-cloning theorem. We assume communication between the adversary's quantum machine and party's quantum machines is perfect: the adversary can simulate any environmental effect or noise on qubits sent by a party.

Queries. The adversary can direct a party to perform certain actions by sending any of the aforementioned activation queries over party's the c and q channels.

The adversary has neither immediate control and cannot observe the content exchanged between the classical and quantum subcomponents of a party over the e channel, nor has information about the bits obtained from the r-channel. Furthermore, to allow for information leakage the adversary may issue the following queries to parties:

- $\mathsf{RevealNext} \rightarrow X$: This query allows the adversary to activate the classical Turing machine to read input from the r-channel and learn future public values. The activated party generates a new value pair (x, X), records it as unused, and returns the public value X. (This query may be specialized if there are multiple value pair types specified by the protocol.)
- $\mathsf{Reveal}(X) \rightarrow x$: This query allows the adversary to compromise secret values used in the protocol computation.[8] If the party has a value pair (x, X) in its memory, it returns the private value x. $\mathsf{Reveal}(\Psi)$ returns the secret key sk for session Ψ, if it exists; this is often referred to as a $\mathsf{RevealSessionKey}$ query.

Where necessary to avoid ambiguity, we use a superscript to indicate the party to whom the query is directed, for example $\mathsf{SendC}^{P_i}(\Psi, msg)$.

Revealing. If (x, X) is a value pair, with public key value or public label X, then the adversary is said to have *revealed the secret for* X if the adversary issued the query $\mathsf{Reveal}(X)$ to a party holding that value pair in its memory. In general, the adversary can reveal the secret for any value X, though this may affect which sessions are fresh.

2.3 Security definition

For the purpose of defining session key security, the adversary has access to the following additional oracle:

- $\mathsf{Test}(i, \Psi) \rightarrow \kappa$: If party P_i has not output a session key, return $\perp$. Otherwise, choose $b \xleftarrow{\$} \{0, 1\}$. If $b = 1$, then return the session key sk from the output for session Ψ at party P_i. If $b = 0$, return a random bit string of length equal to the length of the session key sk in session Ψ at party P_i. Only one call to the Test query is allowed.

Definition 2 (Fresh session). *A session Ψ owned by an honest party P_i is* fresh *if all of the following occur:*

1. *For every vector $\mathbf{v}_j$ in P_i's output for session Ψ, there is at least one element X in $\mathbf{v}_j$ for which the adversary has not revealed the secret.*
2. *The adversary did not issue $\mathsf{Reveal}(\Psi')$ to any honest party P_j for which Ψ' has the same public output vector as Ψ (including the case where $\Psi' = \Psi$ and $P_j = P_i$).*

[8] Our notation here is altered from that of Goldberg et al. [20], in that we call this query Reveal instead of their original term $\mathsf{Partner}$.

3. At the time of session completion, *for every vector* $\mathbf{u}_j$, $j \geq 1$, *in* P_i*'s output for session* Ψ, *there was at least one element* X *in* $\mathbf{u}_j$ *for which the adversary has not revealed the secret.*

The difference between the first condition (involving $\mathbf{v}$) and the third condition (involving $\mathbf{u}$) is that there are some values ($\mathbf{u}$) that are okay for the adversary to learn after the session completes but not before, whereas there may be other values ($\mathbf{v}$) that he can never learn.

Definition 3 (Security). *Let* k *be a security parameter. An authenticated key exchange protocol is* secure *if, for all adversaries* $\mathcal{A}$ *with classical runtime bounded by* $t_c(k)$, *quantum runtime bounded by* $t_q(k)$, *and quantum memory bounded by* $m_q(k)$, *the advantage of* $\mathcal{A}$ *in guessing the bit* b *used in the* Test *query of a fresh session is negligible in* k; *in other words, the probability that* $\mathcal{A}$ *can distinguish the session key of a fresh session from a random string of the same length is negligible.*

Output vectors. One of the key differences between our model and traditional AKE security models is how we phrase restrictions on what secret values the adversary can learn and when. In the eCK model, for example, a fresh session is defined as one in which the adversary has not learned (a) both the session owner's ephemeral secret key x and long-term secret key a, and (b) both the peer's ephemeral secret key y and long-term secret key b (or just the peer's long-term key if no matching peer session exists). In our model, this could be specified as $\mathbf{v} = (\mathbf{v}_0 = (a, x), \mathbf{v}_1 = (b, y))$.

Since in traditional AKE security models the restriction on values learned is specified in the security model, a new security model is required for each differing combination of learnable values. Though models may often appear similar, they sometimes contain subtle but important formal differences and thus become formally incomparable [33]. The traditional approach of specifying the values that can or cannot be learned in the security definition itself contrasts with our approach—building on that of Goldberg et al. [20]—where the vectors $\mathbf{v}$ and $\mathbf{u}$ in the session output specify what can or cannot be learned. As a result, two protocols with differing restrictions on values that can be learned could both be proven secure in our model and then compared based on which values can or cannot be revealed.

2.4 Long-term security

One of the main benefits of quantum key distribution is that it can be secure against unbounded adversaries, but this comes at the cost of being unable to use computationally secure cryptographic primitives such as public key digital signatures for authentication. Definition 3 can be used to analyze QKD when computationally secure cryptographic primitives are used by choosing a $t_c(k)$, $t_q(k)$, and $m_q(k)$ such that the cryptographic primitive is believed secure against

such an adversary. The particular values may be chosen based on known classical algorithms for factoring or discrete logarithms and on present-day limits of quantum devices.

Regardless of the bound on the active adversary, we can still recover a very strong form of long-term security by considering an unbounded quantum Turing machine acting after the protocol has completed. In other words, during the run of the protocol, we assume a bounded adversary as in Definition 3; this bounded active adversary produces some classical and quantum transcript which it provides to the unbounded adversary. This models the real-world scenario of an adversary being somewhat limited by its classical and quantum computing equipment now but later having much more powerful equipment or making an algorithmic breakthrough.

Definition 4 (Long-term security). *An AKE protocol is* long-term secure *if, for all unbounded quantum Turing machines $\mathcal{M}$ acting on a classical and quantum transcript produced by a (bounded) adversary $\mathcal{A}$ in Definition 3, the advantage of $\mathcal{M}$ in guessing the bit b used in the* Test *query of a fresh session is negligible in the security parameter.*

Bounds on devices. If $t_q(k) = m_q(k) = 0$, and Definition 4 is omitted, the model reduces to a classical definition for secure session key establishment. It refines the idea of authentication as the session output can explicitly identify how peers were identified and authenticated. Thus any classical protocol analyzed in [20] can also be analyzed in this model.

This model can be used in conjunction with present limitations of quantum devices. While there are ongoing improvements in controlling quantum systems, at present the number of qubits a device can work with is essentially a small constant compared to classical computers. Thus, using our model with appropriate values of $t_q(k)$ and $m_q(k)$, one can devise efficient protocols that are easy to implement but guarantee unconditional future secrecy. An appropriate assumption on $t_c(k)$—for example that all adversaries with polynomial running time $t_c(k)$ cannot solve a particular hard problem—allow the model to be used as existing classical reductionist security models are used.

Of course, the devices available to the adversary can be made unbounded essentially allowing a complete quantum world. Thus the definitions presented here are suitable for analyzing novel QKD protocols. These alternatives show the wide range of scenarios our definitions incorporate. Due to the unified underlying framework it is easier to compare various protocols and decide which one is the best for the task at hand.

3 BB84

We now turn to the BB84 protocol [5]. We first specify the protocol in the language of the model of Section 2, discuss some aspects of our formulation, and complete the section with a security analysis. Our presentation of BB84

explicitly includes the authentication operations. We choose to focus on authentication using digital signatures, rather than authentication using symmetric key primitives, for several reasons: first, establishment of shared secret keys for authentication is in practice harder than authentic distribution of public keys; and second, the short-term and long-term security properties resulting from the use of public key authentication with QKD are not yet understood.

Definition 5. *Let k be a security parameter. The* BB84 protocol *is defined by having parties responding to activations as follows:*

1. Upon activation $\mathsf{SendC}(\texttt{start}, \texttt{initiator}, B)$ the classical Turing machine A does the following:
   (a) create a new session Ψ^A with peer identifier B;
   (b) read n_1 (random) data bits Ψ^A_{dAB} and n_1 (random) basis bits Ψ^A_{bA} from its r-tape;
   (c) send activation $\mathsf{C2Q}(\Psi^A_{bA}, \Psi^A_{dAB})$ on its e-tape, which indicates that the quantum device should encode each data bit from Ψ^A_{dAB} as $|0\rangle$ or $|1\rangle$ if the corresponding basis bit Ψ^A_{bA} is 0, or as $|+\rangle$ or $|-\rangle$ if the corresponding basis bit Ψ^A_{bA} is 1;
   (d) send activation $\mathsf{SendC}(\Psi^A, \texttt{start}, \texttt{responder}, A)$ on its c-tape to B.
2. Upon activation $\mathsf{SendC}(\Psi^A, \texttt{start}, \texttt{responder}, A)$ the classical Turing machine B does the following:
   (a) create a new session Ψ^B with peer identifier A;
   (b) read n_1 (random) basis bits Ψ^B_{bB} from its r-tape;
   (c) send activation $\mathsf{C2Q}(\Psi^B_{bB})$ on its e-tape, which indicates the quantum device should measure the ith qubit in the $|0\rangle/|1\rangle$ if the ith bit of Ψ^B_{bB} is 0, or in the $|+\rangle/|-\rangle$ basis if ith bit of Ψ^B_{bB} is 1.
3. Upon activation $\mathsf{Q2C}(m)$, the classical Turing machine B does the following:
   (a) set Ψ^B_{dAB} equal to m;
   (b) compute $\sigma \leftarrow \mathsf{Sign}_{pk_B}(\Psi^A, \Psi^B, \Psi^B_{bB}, B)$;
   (c) send activation $\mathsf{SendC}(\Psi^A, \Psi^B, \Psi^B_{bB}, \sigma)$ on its c-tape to A.
4. Upon activation $\mathsf{SendC}(\Psi^A, \Psi^B, \Psi^B_{bB}, \sigma)$, the classical Turing machine A does the following:
   (a) verify σ with pk_B;
   (b) discard all bit positions from Ψ^A_{dAB} for which Ψ^A_{bA} is not equal to Ψ^B_{bB}; assume n_2 such positions remain;
   (c) read n_2 (random) bits Ψ^A_{indAB} from its r-tape; set Ψ^A_{chkAB} to be the substring of Ψ^A_{dAB} for which the bits of Ψ^A_{indAB} are 1, and set Ψ^A_{kAB} to be the substring of Ψ^A_{dAB} for which the bits of Ψ^A_{indAB} are 0; let n_3 denote the length of Ψ^A_{kAB}
   (d) compute $\sigma \leftarrow \mathsf{Sign}_{pk_A}(\Psi^A, \Psi^B, \Psi^A_{bA}, \Psi^A_{indAB}, \Psi^A_{chkAB}, A)$;
   (e) send activation $\mathsf{SendC}(\Psi^A, \Psi^B, \Psi^A_{bA}, \Psi^A_{indAB}, \Psi^A_{chkAB}, \sigma)$ on its c-tape to B.
5. Upon activation $\mathsf{SendC}(\Psi^A, \Psi^B, \Psi^A_{indAB}, \Psi^A_{chkAB}, \sigma)$, the classical Turing machine B does the following:
   (a) verify σ with pk_A;
   (b) discard all bit positions from Ψ^B_{dAB} for which Ψ^A_{bA} is not equal to Ψ^B_{bB}
   (c) set Ψ^B_{chkAB} to be the substring of Ψ^B_{dAB} for which the bits of Ψ^A_{indAB} are 1, and set Ψ^B_{kAB} to be the substring of Ψ^B_{dAB} for which the bits of Ψ^A_{indAB} are 0
   (d) let ϵ be the proportion of bits of Ψ^A_{chkAB} that do not match Ψ^B_{chkAB}; if $\epsilon > 0.061$ then abort;
   (e) compute $\sigma \leftarrow \mathsf{Sign}_{pk_B}(\Psi^A, \Psi^B, \epsilon, B)$;

(f) send activation $\mathsf{SendC}(\Psi^A, \Psi^B, \epsilon, \sigma)$ on its c-tape to A.

6. Upon activation $\mathsf{SendC}(\Psi^A, \Psi^B, \epsilon, \sigma)$, the classical Turing machine A does the following:
   (a) verify σ with pk_B;
   (b) read (random) bits Ψ^A_F from its r-tape to construct a random a 2-universal hash function $F : \{0,1\}^{n_3} \to \{0,1\}^{r'}$ (where $r' = n_3 h(\epsilon) + o(n_3)$) for information reconciliation[9] and compute $F' = F(\Psi^A_{kAB})$;
   (c) read (random) bits $\Psi^A_{P,G}$ from its r-tape to generate a random permutation P on n_3 elements and a 2-universal hash function $G : \{0,1\}^{n_3} \to \{0,1\}^{s'}$ (where $s' = n_3(1 - 3h(\epsilon)) + o(n_3)$) for privacy amplification, respectively; compute $\Psi^A_{skAB} \leftarrow G(P(\Psi^A_{kAB}))$;
   (d) compute $\sigma \leftarrow \mathsf{Sign}_{pk_A}(\Psi^A, \Psi^B, F, F', P, G, A)$;
   (e) send activation $\mathsf{SendC}(\Psi^A, \Psi^B, F, F', P, G, \sigma)$ on its c-tape to B;
   (f) output $(sk = \Psi^A_{skAB}, pid = B, \mathbf{v} = (\mathbf{v}_0 = (\ell(\Psi^A_{dAB})), \mathbf{v}_1 = (\ell(\Psi^A_{bAB})), \mathbf{v}_2 = (\ell(\Psi^B_{dAB})), \mathbf{v}_3 = (\ell(\Psi^B_{bAB})), \mathbf{v}_4 = (\ell(\Psi^A_F)), \mathbf{v}_5 = (\ell(\Psi^A_{P,G}))), \mathbf{u} = (\mathbf{u}_1 = (pk_B)))$ (recall $\ell(\cdot)$ denotes the label describing the corresponding secret value).

7. Upon activation $\mathsf{SendC}(\Psi^A, \Psi^B, F, F', P, G, \sigma)$, the classical Turing machine B does the following:
   (a) verify σ with pk_A;
   (b) use F and F' to correct Ψ^B_{kAB} to $\Psi^B_{kAB'}$;
   (c) compute $\Psi^B_{skAB} \leftarrow G(P(\Psi^B_{kAB'}))$;
   (d) output $(sk = \Psi^B_{skAB}, pid = A, \mathbf{v} = (\mathbf{v}_0 = (\ell(\Psi^A_{dAB})), \mathbf{v}_1 = (\ell(\Psi^A_{bAB})), \mathbf{v}_2 = (\ell(\Psi^B_{dAB})), \mathbf{v}_3 = (\ell(\Psi^B_{bAB})), \mathbf{v}_4 = (\ell(\Psi^A_F)), \mathbf{v}_5 = (\ell(\Psi^A_{P,G})),), \mathbf{u} = (\mathbf{u}_1 = (pk_A)))$.

Remark 1. In the output vector $\mathbf{v}$, the values $\ell(\Psi^A_{bAB})$, $\ell(\Psi^B_{bAB})$, $\ell(\Psi^A_F)$, and $\ell(\Psi^A_{P,G})$ appear as single component vectors. But in step 6(e) the values are broadcast in the clear. This may seem a bit contradictory since, if the adversary has revealed the secret for either of those values (and therefore learns their content), the session is not fresh, but because of the broadcast the adversary *does* in fact learn the values corresponding to the aforementioned labels. The important distinction is *when* the adversary obtains these values, either before or after the protocol commences and measurements are performed. For the adversary to learn these values before parties' measurements, it must reveal the secret for these values, violating session freshness. Learning the values after the session completes is not an issue and the values are given to the adversary "for free", without the need for revealing the secrets.

Remark 2. The output vector $\mathbf{u}$ represents the values which the session owner uses to authenticate its peer. Similar to $\ell(\Psi^A_{bAB})$ the authentication information has to be exclusively available to the alleged peer, but only at the time of protocol execution: they may subsequently be revealed.

Observe that for the BB84 protocol above, Alice's own authentication secret pk_A is not included in her $\mathbf{u}$ or $\mathbf{v}$ vectors. This implies that the protocol is resilient to *key compromise impersonation (KCI) attacks* [35, §2.4.2]: even with Alice's authentication keys no party is able to pretend to be someone other than Alice to Alice.

[9] For details on information reconciliation and privacy amplification, see the full version [34, Appendix A].

3.1 Security of BB84

We now show that the BB84 protocol stated above is a secure (Theorem 1) and long-term-secure (Theorem 2) AKE protocol assuming that the bounded active adversary cannot break the signature scheme.

Theorem 1 (Security of BB84). *Let k be a security parameter. Suppose that the probability ϵ_{sig} that any probabilistic polynomial time classical Turing machine with oracle access to a $(t_q(k), m_q(k))$-bounded quantum Turing machine can break the signature scheme is negligible in k. Then the BB84 protocol is a secure AKE protocol (Definition 3).*

Proof sketch. Our proof combines an existing proof of security by Christandl et al. [36] for the BB84 protocol with the sequence-of-games technique of Shoup [37]. First we show—using techniques from classical reductionist security—that no bounded adversary can (except with negligible probability) successfully tamper with the classical authenticated communication. Then we show—using techniques from QKD security proofs—that the adversary cannot distinguish the key from random. Details appear in the full version [34].

Theorem 2 (Long-term security of BB84). *Let k be a security parameter. Suppose the signature scheme is secure against all bounded adversaries as specified in Theorem 1. Then the BB84 protocol is a long-term secure authenticated key exchange protocol (Definition 4).*

Proof. The argument in fact appears in the proof of Theorem 1. In its proof, the bounds on $t_c(k)$, $t_q(k)$, and $m_q(k)$ and on the adversary are required only for guaranteeing the authenticity and origin of messages in a game hop that assures that the classical authentic communication has not been tampered with. The remainder of the argument is a typical argument for a quantum key distribution scheme, which does not require any bounds on the adversarial power. Since the unbounded adversary runs after the protocol completes, meaning it cannot inject reorder or modify messages in the transcript, therefore the past classical communication remains authentic and the result follows.

4 Comparing classical and quantum key exchange protocols

Given the similarity of our model for both classical and quantum AKE protocols to existing classical AKE security models and our model's flexibility in analyzing the security of a variety of protocols, we can use our model to identify qualitative differences between classes of protocols.

One of the key differences between existing AKE security models such as CK01 and eCK is what randomness the adversary is allowed reveal—and when—yet still have the protocol be secure. Our framework is more generic: it is not the *model* that specifies which randomness can be revealed but the *protocol itself* in

Table 1. Comparison of security properties of various classical and quantum AKE protocols.

Protocol	Signed Diffie–Hellman [2]	UP [23]	BB84 [5]	EPR [6]	BHM96 [7,12]
Protocol type	classical	classical	quantum prepare-send-measure	quantum measure-only	quantum prepare-send-only
Security model	CK01 [2]	eCK [3], this paper	this paper	this paper	this paper
Randomness revealable **before** protocol run?	× static key × ephemeral key	at most 1 of static key, ephemeral key	× static key × basic choice × data bits × info. recon. × priv. amp.	× static key × basis choice × info. recon. × priv. amp.	× static key × basis choice × data bits × info. recon. × priv. amp.
Randomness revealable **after** protocol run?	✓ static key × ephemeral key	at most 1 of static key, ephemeral key	✓ static key ✓ basis choice × data bits ✓ info. recon. ✓ priv. amp.	✓ static key ✓ basis choice ✓ info. recon. ✓ priv. amp.	✓ static key ✓ basis choice × data bits ✓ info. recon. ✓ priv. amp.
Short-term security	computational assumption	computational assumption	computational or inf.-th.	computational or inf.-th.	computational or inf.-th.
Long-term security w/short-term-secure authentication	×	×	✓	✓	✓

its output vectors **v** and **u**. As a result, we can "compare" protocols by viewing them all within our model and then comparing which values are included in the output vector.[10]

Table 1 summarizes the observations of this section. We compare two qualitatively different classical AKE protocols and three qualitatively different QKD protocols: (1) the signed Diffie–Hellman protocol [2] (which can be proven secure in the CK01 model), (2) the UP protocol [23], a variant of the MQV protocol [38] which can be proven secure in the eCK model, (3) the BB84 [5] prepare-send-measure QKD protocol, (4) the EPR [6] (entanglement-based) measure-only QKD protocol, and (5) the BHM96 [7,12] prepare-send-only QKD protocol. Our model is flexible enough to allow all these protocols to be proven secure in it, of course with different cryptographic assumptions, bounds on the adversary, and different output vectors, which we compare in Table 1.

Revealing randomness before the run of the protocol. Some classical AKE protocols, especially eCK-secure protocols such as UP and similar MQV-style protocols, remain secure even if the adversary learns either the ephemeral secret key or the long-term secret key, but not both, before the run of the protocol. This contrasts with all known QKD protocols, where none of the random values—the long-term secret key, the basis choices (for measure protocols), data bits (for prepare protocols), information reconciliation function, or privacy amplification function—can be revealed to the adversary in advance. (This is why all of these values are included individually in the output vector **v** in the BB84 specification in Section 3.)

Revealing randomness after the run of the protocol. For classical AKE protocols to remain secure, at least some secret values must not be revealed after the

[10] We note that it has been shown [33] that the CK01 and eCK models are *formally incomparable*, meaning neither can be shown to imply the other.

run of the protocol. For protocols with so-called perfect forward secrecy, such as signed Diffie–Hellman, the parties' long-term secret keys can be corrupted after the run of the protocol, but not the ephemeral secret keys. For eCK-secure protocols such as MQV-style protocols like `UP`, either the long-term or the ephemeral secret key, but not both, can be revealed before, during, or after the protocol run. For measure-only entanglement-based QKD protocols such as EPR, all random choices made by the parties can be revealed after the run of the protocol: this is because the key bits are not chosen by the parties, nor in fact by the adversary, but are the result of measurements and (after successful privacy amplification) are uncorrelated with any of the input bits of any of the parties, including the adversary. This is not the case for prepare-and-send protocols such as BB84 or BHM96, as the sender randomly chooses data bits which must remain secret.

Short-term and long-term security. Classical AKE protocols can be proven secure only under computational assumptions, and as such only offer short-term security in the sense of Definition 3. Even against an unbounded passive adversary they do not retain any of their secrecy properties. Thus classical AKE protocols are only secure against bounded short-term adversaries; however, they can be compared on the relative strength of the bound on the adversary. This contrasts with QKD protocols. QKD can be shown to be secure against either *unbounded* short-term adversaries, by using information-theoretic authentication, or secure against bounded short-term adversaries when using a computationally secure authentication scheme as we have shown for BB84 in Section 3.1. A key contribution of the model in Section 2 is a formalism which captures the notion that QKD can remain secure against an unbounded adversary after the protocol completes, provided the adversary at the time of the run of the protocol could not break the authentication scheme.

Applications wishing to achieve both long-term security (like QKD) and resistance to randomness revelation (like eCK-secure classical AKE protocols) could do so by running both protocols in parallel for each session, and then combining the keys output by the two protocols together; if combined correctly, the resulting key would provide strong short-term security and strong long-term security. This approach is being used by QKD implementers, such as commercial QKD vendor ID Quantique.[11]

5 Conclusions

We have presented a model for key establishment which incorporates both classical key agreement and quantum key distribution. Our model can accommodate a wide range of practical and theoretical scenarios and can serve as a common framework in which to compare relative security properties of different protocols. A key aspect of our model is that restrictions on values the adversary can compromise are not specified by the model but by the output of the protocol. Using our model, we were able to provide a formal argument for the short-term and

[11] `http://www.idquantique.com/images/stories/PDF/cerberis-encryptor/cerberis-specs.pdf`

long-term security of BB84 in the multi-user setting while using computationally secure authentication.

The ability to compare various classical and quantum protocols in our model has allowed us to identify an important distinction between existing classical and quantum key exchange protocols. At a high level, classical protocols can provide more assurances against online adversaries who can leak or infiltrate in certain ways, but in the long run may be insecure against potential future advances. Current quantum protocols provide assurances against somewhat weaker online adversaries but retain secrecy indefinitely, even against future advances in computing technology.

Since in our model the relative strength of a fresh session is specified by the conditions given in the output vector, an interesting open problem would be to use our model develop a quantum key distribution protocol which does retain its security attributes in the short- and long-terms even if some random values were known before the run of the protocol. Also of interest is how to best combined keys from both quantum and classical key exchange protocols run in parallel.

Acknowledgements

The authors acknowledge helpful discussions with Norbert Lütkenhaus, Alfred Menezes, and Kenny Paterson.

MM is supported by NSERC (Discovery, SPG FREQUENCY, CREATE), QuantumWorks, MITACS, CIFAR, ORF. IQC and Perimeter Institute are supported in part by the Government of Canada and the Province of Ontario.

References

1. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In Stinson, D.R., ed.: Advances in Cryptology – Proc. CRYPTO '93. Volume 773 of LNCS., Springer (1993) 232–249
2. Canetti, R., Krawczyk, H.: Analysis of key-exchange protocols and their use for building secure channels. In Pfitzmann, B., ed.: Advances in Cryptology – Proc. EUROCRYPT 2001. Volume 2045 of LNCS., Springer (2001) 453–474
3. LaMacchia, B., Lauter, K., Mityagin, A.: Stronger security of authenticated key exchange. In Susilo, W., Liu, J.K., Mu, Y., eds.: First International Conference on Provable Security (ProvSec) 2007. Volume 4784 of LNCS., Springer (2007) 1–16
4. Canetti, R.: Universally composable security: a new paradigm for cryptographic protocols (extended abstract). In: Proc. 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS) 2001, IEEE Press (2001) 136–145
5. Bennett, C.H., Brassard, G.: Quantum cryptography: public key distribution and coin tossing. In: Proc. IEEE International Conf. on Computers, Systems and Signal Processing, IEEE (December 1984) 175–179
6. Ekert, A.K.: Quantum cryptography based on Bell's theorem. Physical Review Letters **67** (August 1991) 661–663
7. Biham, E., Huttner, B., Mor, T.: Quantum cryptographic network based on quantum memories. Physical Review A **54**(4) (1996) 2651–2658

8. Mayers, D.: Quantum key distribution and string oblivious transfer in noisy channels. In Koblitz, N., ed.: Advances in Cryptology – Proc. CRYPTO '96. Volume 1109 of LNCS., Springer (1996) 343–357
9. Lo, H.K., Chau, H.F.: Unconditional security of quantum key distribution over arbitrarily long distances. Science **283**(5410) (1999) 2050–2056
10. Biham, E., Boyer, M., Boykin, P.O., Mor, T., Roychowdhury, V.: A proof of the security of quantum key distribution (extended abstract). In: Proc. 32nd Annual ACM Symposium on the Theory of Computing (STOC), ACM Press (2000) 715–724
11. Shor, P., Preskill, J.: Simple proof of security of the BB84 quantum key distribution protocol. Physical Review Letters **85**(2) (2000) 441–444
12. Inamori, H.: Security of practical time-reversed EPR quantum key distribution. Algorithmica **34**(4) (2002) 340–365
13. Gottesman, D., Lo, H.K., Lütkenhaus, N., Preskill, J.: Security of quantum key distribution with imperfect devices. Quantum Information and Computation **4**(5) (September 2004) 325–360
14. Renner, R.: Security of Quantum Key Distribution. PhD thesis, Swiss Federal Institute of Technology Zürich (2005)
15. Paterson, K.G., Piper, F., Schack, R.: Quantum cryptography: A practical information security perspective. In Zukowski, M., Kilin, S., Kowalik, J., eds.: Proc. NATO Advanced Research Workshop on Quantum Communication and Security. Volume 11 of NATO Science for Peace and Security Series, Sub-Series D: Information and Communication Security., IOS Press (2007) See also `http://arxiv.org/abs/quant-ph/0406147`.
16. Alléaume, R., Bouda, J., Branciard, C., Debuisschert, T., Dianati, M., Gisin, N., Godfrey, M., Grangier, P., Länger, T., Leverrier, A., Lütkenhaus, N., Painchault, P., Peev, M., Poppe, A., Pornin, T., Rarity, J., Renner, R., Ribordy, G., Riguidel, M., Salvail, L., Shields, A., Weinfurter, H., Zeilinger, A.: SECOQC white paper on quantum key distribution and cryptography (January 2007) `http://www.arxiv.org/abs/quant-ph/0701168`.
17. Stebila, D., Mosca, M., Lütkenhaus, N.: The case for quantum key distribution. In Sergienki, A., Pascazio, S., Villoresi, P., eds.: Quantum Communication and Quantum Networking: First International Conference, QuantumComm 2009. Volume 36 of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering., Springer (2010) 283–296
18. Ioannou, L.M., Mosca, M.: A new spin on quantum cryptography: Avoiding trapdoors and embracing public keys. In Yang, B.Y., ed.: Proc. 4th International Workshop on Post-Quantum Cryptography (PQCrypto) 2011. Volume 7071 of LNCS., Springer (2011) 255–274
19. Unruh, D.: Everlasting quantum security. Cryptology ePrint Archive, Report 2012/177 (2012) `http://eprint.iacr.org/`.
20. Goldberg, I., Stebila, D., Ustaoglu, B.: Anonymity and one-way authentication in key exchange protocols. Designs, Codes and Cryptography **67**(2) (May 2013) 245–269
21. Cachin, C., Maurer, U.: Unconditional security against memory-bounded adversaries. In Kaliski Jr., B.S., ed.: Advances in Cryptology – Proc. CRYPTO '97. Volume 1297 of LNCS., Springer (1997) 292–306
22. Müller-Quade, J., Unruh, D.: Long-term security and universal composability. Journal of Cryptology **23**(4) (2010) 594–671

23. Ustaoğlu, B.: Comparing SessionStateReveal and EphemeralKeyReveal for Diffie-Hellman protocols. In Pieprzyk, J., Zhang, F., eds.: Provable Security: Third International Conference, ProvSec 2009. Volume 5848 of LNCS., Springer (2009) 183–197
24. Carter, J.L., Wegman, M.N.: Universal classes of hash functions. Journal of Computer and System Sciences **18**(2) (1979) 143–154
25. Wegman, M.N., Carter, J.L.: New hash functions and their use in authentication and set equality. Journal of Computer and System Sciences **22**(3) (1981) 265–279
26. Ben-Or, M., Horodecki, M., Leung, D.W., Mayers, D., Oppenheim, J.: The universal composable security of quantum key distribution. In Kilian, J., ed.: Theory of Cryptography Conference (TCC) 2005. Volume 3378 of LNCS., Springer (2005) 386–406
27. Ben-Or, M., Mayers, D.: General security definition and composability for quantum & classical protocols (2004) arXiv:quant-ph/0409062.
28. Fehr, S., Schaffner, C.: Composing quantum protocols in a classical environment. In Reingold, O., ed.: Theory of Cryptography Conference (TCC) 2009. Volume 5444 of LNCS., Springer (2009) 350–367
29. Unruh, D.: Simulatable security for quantum protocols arXiv:quant-ph/0409125. Extended abstract published as [31].
30. Unruh, D.: Universally composable quantum multi-party computation (full version) (October 2009) arXiv:0910.2912. Short version published as [31].
31. Unruh, D.: Universally composable quantum multi-party computation. In Gilbert, H., ed.: Advances in Cryptology – Proc. EUROCRYPT 2010. Volume 6110 of LNCS., Springer (2010) 486–505 Full version available as [30].
32. Aharonov, D., Ben-Or, M., Eban, E.: Interactive proofs for quantum computations. In Yao, A.C.C., ed.: Proc. Innovations in Computer Science (ICS) 2010. (October 2010) 453–469
33. Cremers, C.: Examining indistinguishability-based security models for key exchange protocols: the case of CK, CK-HMQV, and eCK. In: Proc. 6th ACM Symposium on Information, Computer and Communications Security (ASIACCS) 2011, ACM (2011) 80–91
34. Mosca, M., Stebila, D., Ustaoğlu, B.: Quantum key distribution in the classical authenticated key exchange framework. Cryptology ePrint Archive, Report 2012/361 (2012) `http://eprint.iacr.org/2012/361`, see also `http://arxiv.org/abs/1206.6150`.
35. Boyd, C., Mathuria, A.: Protocols for Authentication and Key Establishment. Springer (2003)
36. Christandl, M., Renner, R., Ekert, A.: A generic security proof for quantum key distribution (February 2004) `http://arxiv.org/abs/quant-ph/0402131v2`.
37. Shoup, V.: Sequences of games: A tool for taming complexity in security proofs. `http://www.shoup.net/papers/games.pdf` (2006) First version appeared in 2004.
38. Law, L., Menezes, A., Qu, M., Solinas, J., Vanstone, S.A.: An efficient protocol for authenticated key agreement. Designs, Codes and Cryptography **28**(2) (2003) 119–134