Certified Quantum Security

Section a. State-of-the-art and objectives

Project in a nutshell. The central objective of this project is to enable formal verification (a.k.a. certification) of *quantum* cryptographic systems, on par with the state of the art in *classical* cryptography. We will develop proof methods, logics for formal verification, and tools for verification of quantum cryptography (both post-quantum cryptography and cryptographic protocols). In parallel, and closely coupled to the research on verification, we will develop new quantum protocols and certify their security. (This has the threefold benefit of being a testbed for our verification technology, an application for it, and a contribution to quantum cryptography in its own right.)

Quantum cryptography. Digital communication permeates all areas of today's daily life. Personal information, business transactions, medical data, financial records, government secrets, all of these are transmitted digitally and need to be protected from attacks against their privacy and integrity. Modern cryptographic systems are used to secure our data.

When (sufficiently universal) quantum computers are built, they will break most of today's cryptography. For example, all cryptosystems using RSA encryption or elliptic curve cryptography will be broken (in particular, all widely commercially deployed public-key encryption schemes). To counter this threat, we need new cryptosystems that withstand quantum attacks ("**post-quantum**" secure protocols), and we need new security proofs for existing ones (that take into account quantum adversaries). Although quantum computers are not available today, post-quantum secure cryptosystems are needed already today: the transition of existing systems can take many years, the development of new systems has to take into account future requirements, and last but not least, highly sensitive data encrypted today might be stored by an attacker and decrypted years from now when quantum computers become available.

On the other hand, quantum technologies introduce new opportunities for the design of secure systems: **quantum protocols** use quantum communication to achieve higher security than is possible classically, or even to perform tasks that are impossible classically. Quantum key distribution (to enable secret communication) is the most well-known such protocol, and already available commercially. Yet the possibilities of quantum protocols go far beyond this single task. For example, we know quantum protocols for quantum position-verification [44, 69] (allowing a device to prove its position in space), and for everlastingly secure multiparty computation [65] (a security paradigm that ensures that no data is leaked even in the light of future technological advances), or quantum money [74, 35]; all these tasks are impossible classically. All of these examples combine methods both from quantum information and from classical cryptography (for dealing with computationally limited adversaries). However, this interplay makes design and analysis complex and error-prone, and careful proofs are needed to ensure security.

The need for verification in cryptography. When designing cryptographic systems, arguably the most important requirement is that there are no attacks that the designer overlooked. (Examples abound where systems were later broken because the designer overlooked a detail.) To achieve this, the state-of-the-art approach is to give mathematical proofs of their security.[1] Unfortunately, security proofs in cryptography tends to be complex, and even small mistakes in cornercases can allow attack vectors to go unnoticed. Since cryptographic proofs are usually checked by humans, such mistakes will often go unnoticed. In a nutshell, the fact that a cryptographic system has been proven secure (and the result published in a reputable venue) does not guarantee the security of the system. This, of course, invalidates the main purpose of the proof, namely to create justified trust in the security of the system.

To solve this problem, computer-aided verification can be used. Here, the security proof is checked and certified by a computer, not by a human. (The proof may still be designed by a human, though.)

[1] An exception is the design of certain elementary primitives such as block ciphers and hash functions, where this approach is usually not pursued because it leads to constructions that are too inefficient.

We distinguish two approaches: verification in the symbolic model, and verification in the computational model. Verification in the symbolic model (pioneered by Dolev and Yao [32]) abstracts from the details of the underlying cryptography. This approach will overlook attacks that make use of those details; the advantage is that this approach can often be automated, and is better suited for the analysis of large protocols. Verification in the computational model avoids such idealizations, modeling explicitly the computations required for breaking the system. This makes the computational model more realistic, but proofs in the computational model become much harder, and can rarely be fully automated. In this proposal, we focus on verification in the computational model.

Certified, computer-verified proofs in the computational model enable highest trust in cryptographic systems, beyond what can be achieved by human inspection of security proofs. I believe that such a high level of trust is needed when using cryptography in high security settings, or when standardizing cryptographic systems for widespread use.

The need for quantum verification. When designing quantum cryptographic systems – be it post-quantum secure classical protocols, or quantum protocols in the computational model – to ensure security against quantum attackers we need to: (a) base new cryptographic schemes on computational problems that are supposedly hard for quantum computers (such as certain lattice-problems, but not factoring of integers or discrete logarithms) and (b) perform the security proofs in a computational model that includes quantum computation. Depending on the scheme, the latter can add considerable challenges since quantum security proofs can be considerably more complex than their classical counterparts, and also because they require an additional concentration of know-how: methods both from cryptography and from quantum computing/quantum information need to be used in the same proof. Also, proofs that take into account quantum mechanics may be more likely to contain errors, since human intuition has evolved to understand the classical world, but not quantum physics.

With those facts in mind, we argue that verification should be applied not just to classical schemes that are secure against classical attackers. Instead, approaches to the verification of security should take into account attackers that have a quantum computer (and quantum communication). In a nutshell, verification tools should model the adversary as quantum, and preferably also allow us to analyze protocols that explicitly make use of quantum technologies for honest purposes (e.g., quantum key distribution). Such reasoning tools will need considerably different underlying logics that capture the peculiarities of quantum mechanics.

Objectives. Our main goal is:

> **Main objective:** To develop techniques and tools for the certification of quantum protocols, applicable to real-world quantum cryptographic systems.

This includes the development of logics for reasoning about quantum protocols (both in the case of post-quantum cryptography and of quantum protocols), usable tools for performing the verification on a computer, and case studies that show the applicability and scalability of our methods. This objective is addressed by WP1–WP3.

In addition, we pursue the following secondary goal:

> **Secondary objective:** To develop post-quantum secure cryptographic schemes to replace insecure classical schemes, and to develop quantum protocols for new (classically impossible) problems. To develop the proof techniques required to show the security of these schemes. And to formally certify the security of these schemes.

For quantum protocols, we focus especially on constructions that combine methods from classical cryptography and quantum information (i.e., protocols with computational security), e.g., computational quantum commitments or obfuscation. The goal is to map the possibilities that quantum technologies can bring in the future.

The secondary objective is addressed in WP4 (for the development of the schemes) and WP3 Task 6 (for their verification).

The secondary objective, besides advancing the state-of-the-art of quantum cryptography on its own (new quantum protocols with certified security), ties in directly into the main objective: Development of formal verification techniques goes hand-in hand with the development of the proof

techniques used in pen-and-paper proofs. The analysis of full-fledged schemes tests the applicability of our verification methods for real-world cryptographic protocols. Lessons learned during verification can guide the protocol design process and lead to "verification friendly" protocols.

Final result. In the end of the project, we plan to have demonstrated the whole pipeline of developing a certified quantum protocol: from the design of a new protocol, through pen-and-paper security proof, till formal certification in a theorem prover such as Isabelle/HOL.

State of the art. Existing research on verification of cryptography focuses on classical cryptography. Most notable examples are the tools CryptoVerif [14, 27], CertiCrypt [10, 21], and EasyCrypt [9, 33] (as well as the logics underlying these systems). EasyCrypt and CertiCrypt use a probabilistic relational Hoare logic (pRHL) that allows us to reason about the equivalence of pairs of probabilistic "programs" (representing executions of cryptographic schemes), while CryptoVerif performs a (partially) automated rewriting of programs. These tools perform a computer-aided verification of *classical* cryptographic schemes. That is, the underlying logics implicitly assume that the adversary is classical. Can the same tools be used for the analysis of quantum-secure schemes? For EasyCrypt, we explicitly showed that this is not the case: We constructed an example [62] where EasyCrypt asserts security, but which is known to be insecure against a quantum attacker. This shows that the pRHL underlying EasyCrypt (and CertiCrypt) is not suitable for analyzing quantum-secure protocols (and even less for protocols that use quantum communication). Currently, no tools for verifying cryptographic protocols exist that take into account quantum attackers in the computational model. Nor have the theoretical foundations for such tools been developed.

There is a wealth of results on logics for the verification of quantum programs. Quantum programs are closely connected to quantum protocols because (especially when analysing protocols in the computational model) quantum protocols are modeled as small program fragments, called "games". For example, Hoare logics and weakest precondition calculi have been presented by D'Hondt and Panangaden [31], Chadha, Mateus and Sernadas [22], Feng, Duan, Ji, and Ying [37], and Ying [76], to cite only a few. (The area is too big to summarize in this overview. See, e.g., the book by Ying [77] for more references.) These logics provide us with reasoning tools to derive deterministic and probabilistic properties of quantum programs. Yet, in the analysis of cryptographic protocols, we typically need to show that two programs stand in a certain relationship; none of the aforementioned logics supports this.

Existing approaches for showing the equivalence of quantum programs focus either on the explicit calculation of the quantum state resulting from an execution of the programs, e.g., [6, 7], or on bisimilarity of quantum process calculi, e.g., [46, 47, 38], or on categorical diagrams [1, 25, 45]. Notice that the explicit calculation will work only for finite (and relatively small) systems, and is thus inapplicable to computational cryptography (which has necessarily an exponentially large state space). Bisimilarity can show the exact equivalence between two programs/processes, but it cannot model computationally-limited adversaries,[2] and they do not allow us to describe the relationship between processes in a more refined way (like "program B's output is shorter than program A's output if condition C holds"; pRHL excels at these kinds of statements). Judging from the experience in verification of classical cryptography, the verification of bisimilarity and related notions is a powerful tool for the analysis in the symbolic model (cf., e.g., the tools Proverif [15, 53] and Adecs [23, 3]), but lacks applications in the computational model. (In fact, when CertiCrypt [10] was first developed, the situation in the classical case was similar to the current situation in the quantum case. The literature contained an abundance of logics for the analysis of deterministic and probabilistic programs. Yet, to meaningfully compare two probabilistic programs in the computational setting, a new logic needed to be developed, namely pRHL.)

[2] Note that with respect to unlimited adversaries, some cryptographically relevant results have been achieved. Several papers analyze the security of the BB84 [12] quantum key distribution (QKD) protocol. [38, 57] show security against a very simple, specific intercept-resend attack (that measures all qubits in a random basis), [46, 47] verify one step of the Shor-Preskill proof [54] for QKD. [45] shows security of a toy variant of BB84, though only secure against intercept-resend attacks and covert adversaries tolerating only *exponentially small* error. It is not clear that these approaches can scale to the full proof of QKD. But they may be useful in combination with the techniques developed in this project to automate specific kinds of substeps in larger security proofs, see WP1 Task 4.

Closely related to the verification of quantum programs is also research on the semantics of quantum programs and of quantum processes. While semantics for basic languages (such as imperative languages without higher-order functions or quantum control[3]) are straightforward, formalizing more advanced language features is challenging. [4] were the first to present a language with quantum control, with more recent advances in [78, 19]. Semantics of higher-order functions were given by, e.g., [52]. A large part of this research consists of identifying suitable categories in which the semantics of quantum programs (or more generally quantum processes) can be expressed, starting with the seminal work of Abramsky and Coecke [1] on category-theoretical modeling of quantum processes. Much progress was made in this area since, see for example the survey [25] for advances along the line of the Abramsky-Coecke approach, and [24] for a survey of Jacobs' related effectus theory [42] (the latter has the convenient feature that it unifies the concept of effects (essentially, programs) and predicates). This research on semantics of quantum programs is relevant but not central for our project since in most situations, simple imperative quantum languages are sufficient for expressing cryptographic games. (But see WP1 Task 6.)

To summarize, both the area of verification of classical cryptography and of verification of quantum programs are well studied. However, for machine-aided verification of *computational* quantum cryptography, neither theoretical foundations nor tools exist so far.

This gives an overview over the state of the art for verification of quantum protocols. The state of the art for quantum cryptography in general is too wide to summarize here. However, for the problems tackled in WP4 (which deals with the design and analysis of quantum cryptographic protocols), we give additional specific background there.

Section b. Methodology

EasyCrypt [9, 33], together with its underlying probabilistic relational Hoare logic (pRHL; [10]) seems to be the most successful approach today for formally verifying cryptography in the computational model.[4] The main hypothesis underlying our research proposal is:

> Although EasyCrypt and its underlying logic pRHL are not sound for quantum adversaries, the basic ideas underlying EasyCrypt/pRHL are well-suited also for the quantum case. (Assuming a new logic with similar intuitive meaning but different formal semantics.)

This hypothesis is supported by the fact that hand-written cryptographic proofs in the classical and quantum setting follow similar patterns (even though the techniques may vary very much in some details): The most rigorous hand-written proofs in the classical setting are usually formalized as game-based proofs (a.k.a. game-hopping proofs) [55, 11]; the PI of this project has successfully used the same formalism for the analysis of quantum protocols that involve computational cryptography (e.g., [68, 28, 56, 72, 67, 69, 70]). EasyCrypt/pRHL are essentially an attempt to formalize such game-based proofs.

In our project, we will therefore follow the successful strategy of EasyCrypt as far as possible, leveraging existing research where feasible, while developing new technologies where it becomes necessary due to the special nature of quantum mechanics.

Towards this goal, our research will be structured along the following four work packages:

- **WP1. Logical foundations.** We develop the theoretical foundations for reasoning about quantum cryptography. This includes the development of a logic "qRHL" for representing and proving facts about quantum cryptographic schemes, and the development of reasoning rules and automatic decision procedures.

[3] Quantum control means that conditionals can depend on quantum data and execute both branches in superposition.

[4] It is hard to give formal metrics for the success of an approach, but the attention by the community can be used as an indication. The two papers [10, 9] that present the core foundations of the approach have together 413 citations (Google Scholar, 2018-02-12). At least three graduate schools that have a large focus on EasyCrypt have been organized (UPenn EasyCrypt summer school 2013, Joint EasyCrypt/F*/CryptoVerif School 2014, IACR School on Computer-aided Cryptography 2015).

- **WP2. Tool support.** We develop tools (software) for interactively verifying the security of cryptographic protocols (with as much automation as possible). The final tool will allow us to certify the security of a protocol in the theorem prover Isabelle/HOL.
- **WP3. Case studies.** To validate our approach, we perform case studies in which we analyze increasingly complex quantum cryptographic protocols, culminating in the verification of real-world protocols (both post-quantum secure ones, and quantum protocols).
- **WP4. Quantum cryptography.** We develop new post-quantum secure schemes and quantum protocols for existing and emerging applications. These schemes form the basis for the advanced case studies and demonstrate the feasibility of the overall development process of a certified secure protocol: from initial protocol idea till final verification.

These work packages are described in detail below.

Work package 1: Logical foundations

The main underpinning of the project is the development of a quantum-analogue of the pRHL used in EasyCrypt (a "quantum relational Hoare logic", qRHL). In pRHL, we express the relationship between two programs by judgements $\{A\}c \sim d\{B\}$ that express (roughly speaking) that

> if two programs c, d start out in states that are related by the invariant A,
> then the states after execution will be related by the invariant B.

Then, most of the security proof is done by reasoning about such pRHL judgements. The invariants A and B are, in this case, simply formulas that express the relationship between the classical values of the variables used by the programs c and d.

For deterministic programs c, d, defining $\{A\}c \sim d\{B\}$ is straightforward: A is interpreted as a relation on memory contents, and $\{A\}c \sim d\{B\}$ means: For any memory contents $(m_1, m_2) \in A$, we have $(m_1', m_2') \in B$ where m_1' is the memory after running c on memory m_1, and m_2' analogous. For randomized c, d, the definition already becomes non-trivial [10].[5]

In the quantum case, things are even more complicated. Invariants A and B may refer to quantum variables (at the very least, the adversary's state will be a quantum variable). Thus we need to be able to express relations between quantum variables. Even in the simplest proofs (i.e., those which refer to quantum adversaries but are essentially classical) we will need to express that the adversary state is the same in the programs c and d, see below. (This is an invariant that occurs in basically every proof step in an EasyCrypt proof.) We thus need to develop a formalism for stating relations between quantum states. This goes beyond existing quantum Hoare logics such as [22] because those cannot express the relation between quantum registers in different quantum states. In addition, after defining the semantics of invariants A, B, we need to define the semantics of Hoare judgements $\{A\}c \sim d\{B\}$. Our basic approach for solving these problems is: We define predicates A, B to be subspaces of the state space of two possibly entangled memories (the memories of c, d).[6] This allows us to encode the relationship between two memories. By following the ideas behind the formalization of pRHL, we then get the following informal definition for $\{A\}c \sim d\{B\}$:

Definition 1 (Informal) *If ρ_1, ρ_2 describe the quantum state of memories M_1, M_2, respectively, then M_1, M_2 are "related by A" iff there exists a density operator ρ (the "witness" for A) with support in the space A, and with marginals ρ_1, ρ_2. $\{A\}c \sim d\{B\}$ means that if M_1, M_2 are related by A, and M_1', M_2' are the memories after execution of c and d, respectively, then M_1', M_2' are related by B.*

Unfortunately, it turns out that this definition is not very usable. Namely, with this definition we cannot show the following "frame rule":

$$\frac{\{A\}c \sim d\{B\} \qquad \text{variables of } R \text{ and of } A, c, d, B \text{ are disjoint}}{\{A \wedge R\}c \sim d\{B \wedge R\}}$$

[5] For example, the post-condition $x_c \geq x_d$, referring to variables x_c, x_d in c, d, respectively, does not mean that x_c will be greater-equal to x_d, but that the marginal random variable x_c majorizes the random variable x_d.

[6] A possible generalization, but more complicated to use, would be predicates in the style of D'Hondt-Panagaden [31] (i.e., positive operators). However, as a first step we stick to "strict" predicates, i.e., subspaces, leaving more general predicates as a second step.

Such a frame rule is essential for local reasoning about subprograms, allowing us to analyze c, d with respect to the variables occurring in them, and then to generalize to a larger context. A strengthening of Definition 1 ("uniform qRHL") would be to require that the witness for A depends uniformly on the witness for B (namely, that the dependence is described by a superoperator). Unfortunately, while this definition can be shown to satisfy the frame rule, it lacks another important rule, the "equality rule":

$$\frac{X \text{ are the free variables of } c}{\{X_1 = X_2\}c \sim c\{X_1 = X_2\}}$$

Here X_1, X_2 refer to the variables of the left/right instance of c, respectively. This rule intuitively guarantees that if the initial states are the same in two systems, and we apply the same program in both systems, then the states are the same afterwards. Obviously, we would want this to hold (and experiences with EasyCrypt show that such a rule is crucial when reasoning about adversaries). Unfortunately, uniform qRHL does not satisfy that rule. (Note that we have not yet specified what $X_1 = X_2$ formally means. But we can show that there is no interpretation of $X_1 = X_2$ that makes the equality rule true.)

Fortunately, there is another variant of qRHL that satisfies both the frame rule and the equality rule. Namely, if we change Definition 1 to additionally require that the witness is a separable state, then we can show the frame rule. Do we have the equality rule? For this, we need to give a suitable definition of the equality $X_1 = X_2$ (recall, predicates are subspaces). Namely, if we define the predicate $X_1 = X_2$ as the subspace invariant under swapping X_1 and X_2, then the equality rule holds as well.

These example illustrate that there are many design choices when formalizing qRHL, and one of the main challenges is to choose all those details in a way that the resulting logic satisfies all expected derivation rules, and as many useful additional rules as possible.

Note that in preliminary work [71], we already developed a basic qRHL with a rudimentary set of rules. So Task 1 will focus on improving on the basic definition (e.g., to support predicates that can express classicality or non-entanglement of variables), and on developing a richer set of rules.

Task 1: Development of qRHL. *We will develop a formalism for invariants for pairs of memories, and semantics for quantum relational Hoare judgements (qRHL) $\{A\}c \sim d\{B\}$ for programs c, d expressed in a simple imperative quantum programming language qWhile (an extension of the pWhile language [10], with quantum variables but without quantum control flow). We develop a set of sound reasoning rules for qRHL (comparable to the rules used in EasyCrypt). A design goal will be to formulate the invariants and reasoning rules in a way such that, at least when reasoning about classical values, the resulting judgments will look as similar as possible to those used in pRHL (to reduce the learning curve when coming from EasyCrypt, and to make it easier to port proofs from the classical setting). (We do not try to find a* complete *set of rules; even for pRHL no such set is known. This does not limit the practical usefulness of those rules.)*

The reasoning rules developed in Task 1 will be sufficient for analyzing classical cryptographic protocols where the proof of post-quantum security is essentially the same as the one for classical security, and for simple quantum protocols (such as, e.g., teleportation and quantum one-time pads). Security proofs that go beyond this will often need reasoning methods that go beyond those rules. For example, in security proofs involving quantum rewinding,[7] we typically need a reasoning step such as: "If there is an adversary attacking our protocol, then there exists an adversary that consists of an application of a unitary transformation U and that has the same success probability." (This step is useful because afterwards we can use the inverse $U^\dagger$ to undo adversary actions.) In addition, specific lemmas have been developed to handle a number of special situations in quantum cryptographic proofs: For proofs involving rewinding, we have Watrous' rewinding technique [73] (for zero-knowledge proofs) and Unruh's rewinding techniques [70]. For reasoning about random oracles, we have, e.g., Zhandry's lemma about the pseudorandomness of $2q$-wise functions [79], Zhandry's semiconstant distributions [79], Unruh's one-way-to-hiding lemma for programming the random oracle [72]. To reason about

[7] Rewinding is a proof technique used in particular when dealing with zero-knowledge proofs. Rewinding resets the adversary to an earlier state, this technique is challenging to use in the quantum setting because we cannot copy the state for restoring it later.

the knowledge of the adversary in a quantum setting, we often use tools from quantum-information theory such as the quantum leftover hash lemma [60], higher-order uncertainty relations [29], lemmas on quantum sampling [17], to mention a few. To make these tools usable in as many formal proofs as possible, we need to identify commonly used techniques (including the above), find their most general form, and identify how to best express them in our logic (or extensions thereof).[8]

Task 2: Advanced proof tools. *Identify commonly used advanced proof techniques in security proofs (with a focus on rewinding, random oracles, and quantum information). Design sound and general reasoning rules for qRHL that allow us to use those proof techniques in a formal proof. Describe common proof patterns how to use those rules.*

Although classical security does not necessarily imply post-quantum security, in specific cases one can deduce post-quantum security from classical security. Roughly speaking, if classical security holds even with respect to adversaries that have access to arbitrary (interactive and stateful) oracles, then security also holds with respect to adversaries that have access to a quantum computer. Thus we expect that in some cases, a security proof (or parts of it) can be formalized in a classical setting (using pRHL), and then post-quantum security (in qRHL) can be derived by the application of a single transfer rule. This strongly reduces the effort needed to port classical EasyCrypt/pRHL proofs into qRHL (especially when both logics are encoded in a single tool, see WP2 Task 2).

Task 3: Linking pRHL / qRHL. *Determine conditions under which a security property expressed with respect to classical adversaries (using pRHL) implies the corresponding security property with respect to quantum adversaries (using qRHL). Design appropriate proof rules for transferring pRHL statements into qRHL statements.*

pRHL is a useful logic for manually creating a computer-verified proof. But it does not seem to be the best logic for automated verification and decision procedures. More powerful automated tools exist, for example, for deciding equivalence relations between programs, e.g., for bisimilarity or trace equivalence (e.g., [23]). We assume that similarly, qRHL is not the ideal logic for automated reasoning tools. However, some progress has been made towards automatically deciding equivalences of quantum programs in other logics (for programs with a small finite number of steps). For example, [7] successfully analyses small quantum protocols such as teleportation, and [47] can even automatically show a specific proof step in the security proof of the BB84 quantum key distribution protocol. Research initiated by [1] (see [25] for a survey) demonstrates how to use categorical diagrams for proving the equality of quantum programs/protocols, e.g., [45] uses categorical diagrams for proving the equality of diagrams. (To the best of our knowledge, no automated tools for deciding the equivalence of categorical diagrams have been developed so far, though.) Using those tools to automatically prove certain substeps in a qRHL proof would strongly simplify the analysis of many protocols (e.g., we expect that the step automated by [47] would involve much work when done manually in qRHL). To be able to use those tools in a qRHL-based proof, we will need proof rules that say "if c, d are equivalent with respect to the logic of tool X, then $\{A\}c \sim d\{B\}$ holds for suitable pre- and post-conditions A, B".

Task 4: Linking qRHL to other quantum logics. *For existing results for automated decision of equivalences of quantum programs: Identify conditions under which an equivalence as proven by those tools (e.g., a bisimilarity) implies a qRHL judgement. Formalize rules for transferring equivalences into qRHL statements. Identify common proof patterns how to use these rules.*

Task 4 tackles the problem of automation by connecting qRHL to *existing* results for automating equivalence testing. In addition to relying on existing tools, we will develop new decision procedures for suitable classes of qRHL judgements. (Either by reasoning directly on qRHL or by first reducing qRHL to suitable other logics as in Task 4 and developing decision procedures for these.) From the case studies in WP3, we gain experience which kinds of qRHL judgements are important to automate

[8] In particular, lemmas that are commonly expressed in terms of entropies will not necessarily be expressed in a straightforward way in our logic. Finding the right formulation will later simplify the reasoning process considerably.

in real-life proofs, so we can develop decision procedures specifically for those cases (as opposed to Task 4, where we use only existing, generic results).

Task 5: Automation/decision procedures. *Develop decision procedures (or at least sound but incomplete automation procedures) for important classes of subgoals occurring in qRHL-based proofs. Identify common proof patterns how to use these rules.*

The qRHL developed in Task 1 focuses on an imperative language with denotational semantics described in terms of superoperators (completely positive trace-preserving maps) The advantage is that the semantics of such languages are very well understood. However, to use more advanced language features such as higher-order quantum programs or quantum control, we need to use denotational semantics that can express such features, e.g., [52] for higher-order, and [4, 78, 19] for quantum control. (We stress that we do not plan to develop new semantic models in this task. Rather, the focus is to use state-of-the-art semantic models and to adopt qRHL to those.) Furthermore, based on categorical descriptions of predicates and effects (e.g., using effectuses, see [24] for a survey) we may get a more uniform notion of qRHL that unifies the notion of predicates and programs. (E.g., [2] makes this connection explicit by describing a unified language for predicates and programs.)

Task 6: qRHL for advanced languages. *Generalize qRHL to more descriptive semantic models (encompassing quantum control and higher-order programs). Explore more generic models for predicates, e.g., using effectuses or related structures.*

Work package 2: Tool support

WP1 develops the underlying formalism for formal security proofs in the computational model. However, a manual application of the reasoning rules from WP1 without computer support would be very tedious or even infeasible for all but the smallest protocols. In addition, this would be contrary to our main goal, namely to avoid the potential for human error in security proofs. Thus, in this work package, we will develop software tools that help the user to develop and check the proofs.

The first step will be the development of a prototype tool early in the project. This tool will intentionally be kept minimal in its scope. We will provide support for applying the basic rules from WP1 Task 1. This tool will not be optimized for usability. The purpose of the tool is to provide an early validation of our approach, and to serve as a test-bed for exploring which rules are needed for simple crypto proofs.

A first version of this tool was already implemented in our preliminary work [71]. However, we will have to extend the tool continuously to keep up with our progress in Task 1.

Task 1: Prototype tool. *Design a simple tool for checking proofs. The tool will support goal-driven reasoning with qRHL statements as goals, and provide a tactic for each reasoning rule from WP1 Task 1. Subgoals that are not qRHL statements will be outsourced to an SMT solver.*

To get a tool that is usable for the analysis of larger programs, we could extend the tool from Task 1 to increase its usability, and the power of the various tactics. However, this would mean a considerable duplication of effort with respect to existing tools such as EasyCrypt: A large part of EasyCrypt deals with reasoning in first-order logic (not specific to the pRHL judgements), with the module system for specifying more complex protocols and adversaries, and with general usability improvements (e.g., Proof General [59] support). In order to avoid this duplication, we plan to integrate qRHL as an additional logic in the existing EasyCrypt sources (EasyCrypt is open source under the CeCILL-C license), resulting in a QuEasyCrypt (quantum EasyCrypt) tool. Besides avoiding unnecessary effort, this has additional advantages:

- Users experienced with EasyCrypt will find the transition to QuEasyCrypt easier.
- Being able to represent both pRHL and qRHL in a single proof script allows us to easily implement proof techniques from WP1 Task 3 (linking pRHL / qRHL).
- By keeping syntactic differences minimal it will be easier to extend EasyCrypt proofs to the quantum setting.

Task 2: EasyCrypt integration. *Extend the EasyCrypt tool to support qRHL judgements. Implement tactics for all qRHL rules developed in WP1 Tasks 1–3. (For Task 1, this will mostly mean adapting existing EasyCrypt tactics to generate somewhat different subgoals in the qRHL case.) We expect to collaborate closely with the developers of EasyCrypt on this task (Gilles Barthe's group at IMDEA Madrid).*

In WP1 Task 4 we develop rules that link existing tools for equivalence checking to qRHL. In WP1 Task 5 we develop further decision procedures. In order to make use of these results, we also need tool support for these:

Task 3: Automation. *Develop tools for automated decision of important classes of qRHL judgements. Either by translating qRHL statements into the logics covered in WP1 Task 4 for further processing with third-party tools, or by directly implementing the decision procedures from WP1 Task 5. For added convenience, our tools will be integrated into QuEasyCrypt as tactics and automatically invoke the decision procedures and third-party tools.*

The tasks in this work package so far follow the approach used in EasyCrypt as well: User error in the verification is avoided by providing the user with a set of sound reasoning rules. As long as these rules are indeed sound, no incorrect proof is possible. However, the rules themselves might be unsound – their soundness is proven only using pen-and-paper proofs. Additionally, implementation errors can lead to unsound proofs as well. For example, while working with EasyCrypt, we found at least three possibilities of proving false [34].[9] To avoid this, we need to break down the machine-verified proof into a sequence of elementary reasoning steps in an underlying formalization of mathematics (e.g., first-order logic, higher-order logic, etc.) that can be verified by a simple checker (small trusted core). Popular examples of such "LCF-style" theorem provers are Isabelle/HOL [51] and Coq [58]. For example, CertiCrypt [21], the predecessor of EasyCrypt, formalized all proofs in Coq. Similarly, we will formalize qRHL and the tactics and proof methods developed here in Isabelle/HOL. (In prior work, we have already formalized pRHL and some of EasyCrypt's tactics in Isabelle/HOL [66].) An additional advantage is that we inherit the very mature Isabelle framework, which includes a powerful user interface, programmability on the user level, and a much better support for reasoning in the ambient logic than EasyCrypt (i.e., when reasoning about subgoals that do not include qRHL statements).

Task 4: Verification in Isabelle/HOL. *Formalize qRHL in Isabelle/HOL. Prove the soundness of the reasoning rules developed in WP1 in Isabelle/HOL. Implement those rules as tactics in Isabelle/HOL (in close correspondence to the EasyCrypt tactics to make the porting of proofs easier).*

Work package 3: Case studies

The first two work packages develop foundations and tools for the verification of quantum-secure protocols. To validate the usefulness of these tools, we conduct a number of case studies: We formally prove the quantum security of a number of protocols using our methods. This will give us feedback as to which additional reasoning steps should be supported, helps us understand the scalability of our appraoch, yields example proofs that help future researchers to use our tools, and – for the more advanced examples – has the direct benefit of certifying the quantum security of real-world protocols. We expect a mutual dependence between the case studies and WP1/WP2 since experiences during our case studies will lead to improvements of the tools and their underlying theory. This work package culminates in the verification of the protocols we develop in WP4.

Task 1: Toy protocols. *In the first phase of our project, we prove the security of very simple toy protocols. (E.g., one-time pad, composition of two one-way permutations (OWP) is an OWP, or toy protocols made up specifically for this task.) The goal of these toy protocols is to provide simple test cases for each of the reasoning rules, and to test the tool from WP2 Task 1, and to provide test cases for individual tactics while developing QuEasyCrypt (WP2 Task 2).*

[9] This is not a reflection on the quality of EasyCrypt, especially since EasyCrypt is still considered to be in beta-stage. However, it illustrates the danger of relying on the complex implementation of a particular prover.

Task 2: EasyCrypt proofs. *Once QuEasyCrypt (WP2 Task 2) is in a workable state, we will translate the security proofs contributed with EasyCrypt to the quantum setting.*[10] *This will give us feedback on the usability of our approach and of QuEasyCrypt. It will also show how much extra effort is involved in doing a post-quantum security proof (our hypothesis is that the extra effort is minimal for most protocols compared with a classical proof in EasyCrypt). Finally, since the proofs contributed with EasyCrypt are real-world crypto protocols, this also provides additional trust in the security of those schemes.*

Task 3: Rewinding and random oracles. *To validate the techniques developed in WP1 Task 2, we will formally verify a real-life protocol involving rewinding, and a real-life protocol involving random oracles. The choice of the protocols will be decided upon review of the available candidates when the task is started, but we list possible candidates here: For random oracles, full domain-hash (post-quantum security analyzed in [79]), Fujisaki-Okamoto (analyzed in [56]), signatures from sigma-protocols (constructed and analyzed in [67]). For rewinding: a lattice-based identification protocol such as, e.g., [49].*

Task 4: QKD protocol. *To validate the possibility of analysing "truly quantum" protocols, we formally verify the security of a state-of-the-art QKD protocol. (We chose the exact protocol when we begin the task.) We use automation techniques from WP2 Task 3 when possible. In addition, we verify security in the case where computationally-secure signatures are used for implementing the authenticated channel in the QKD protocol. (This combination is particularly relevant, because it improves the usability of QKD without sacrificing what is known as "everlasting security" [65], and at the same time it is a test case for the analysis of protocols involving both computational adversaries and methods from quantum information.)*

Task 5: Proofs in Isabelle/HOL. *When our Isabelle/HOL framework (WP2 Task 4) is ready, we port one or more of the above proofs to Isabelle. (Since the exact effort of porting to Isabelle is hard to judge at this point, we do not try to predict how many of the proofs we will port.)*

Task 6: Protocols from WP4. *We formally verify the security of the protocols and schemes we develop in WP4. This will take place either in QuEasyCrypt or Isabelle, depending on the maturity of the respective implementations.*

Work package 4: Quantum cryptography

In addition to the research directly concerning the *verification* of quantum cryptography we will also develop new cryptographic schemes, both classical schemes secure against quantum adversaries (post-quantum cryptography), as well as quantum protocols (using quantum communication and potentially quantum computing). The security of all schemes developed in this work package will be certified (in WP3 Task 6) using the tools from WP2. Thus the present work package serves a dual purpose: On the one hand, we develop the first certified post-quantum secure schemes and quantum-protocols. On the other hand, the development of protocols and proof techniques is a driving force and gives feedback to our work on verification – what proof techniques are required, how well do our approaches scale, what needs to be improved?

The nature of the present work package is explorative: what protocols can we construct using the techniques at hand in quantum cryptography (and of course, using new techniques beyond the state of the art), what added benefits can quantum communication give us? We list a number of promising research tasks below, but it is to be understood that research on the design and analysis of other post-quantum secure schemes and quantum protocols fall within the scope of this work package as well, as long as the security proofs will be verified formally. Depending on future advances, and on our discoveries, we may deviate from the tasks below and explore other schemes and protocols if they are more promising.

[10] Obviously, only where this makes sense. If a protocol is simply not quantum secure, we cannot translate the proof. If a protocol is quantum secure, but the quantum security proof is very different from the classical one, we also consider the proof out of the scope of this task.

A fundamental primitive in cryptography are hash functions. We showed in [64] that classical security notions for hash functions (collision resistance) are, in many cases, not sufficient in the quantum case. Instead stronger properties such as "collapsing" hashes [64] are needed to get post-quantum security. Constructions exist [64, 63], but they either use random oracles or use strong computational assumptions (hardness of the learning with errors problem LWE).

Task 1: Hash functions. *Construct more efficient construction for collapsing hash functions in the standard model. Prove the security of existing constructions (e.g., SHA3). Identify and formalize further required security properties for post-quantum secure hash functions (besides collapsing).*

A second important building block in cryptography are signatures. Some of the most efficient signature schemes are based on the Fiat-Shamir transformation [39] in the random oracle model. Yet, their post-quantum security is still unclear: in the post-quantum setting, the adversary gets superposition access to random oracles [16] which breaks classical security proofs.

Task 2: Signature schemes / non-interactive zero-knowledge proofs. *Prove the security of the Fiat-Shamir construction [39] (without the strong additional restrictions of our prior work [68]). Develop efficient post-quantum secure alternatives for Fiat-Shamir (more efficient than our prior work [67]). Develop non-interactive zero-knowledge proof systems in the random oracle model. (The latter is technically very closely related to signatures. E.g., both [39] and [67] are simultaneously signatures and proof systems.)*

A third central building block are commitments. We showed in [5] that classical security definitions for commitments are not sufficient to guarantee post-quantum security. Instead, a commitment should be "collapse-binding" [64]. (Roughly speaking, this implies that the adversary cannot open a commitment to several values in superposition.) However, (statistically hiding) collapse binding commitments exist only under strong assumptions (e.g., hardness of LWE or random oracles). Do collapse-binding commitments exist given weak assumptions such as one-way functions?

If we implement commitments using quantum protocols (i.e., leaving the domain of post-quantum cryptography), can we get more efficient protocols than classically? For example, [75] constructs non-interactive statistically-binding commitments from one-way functions (while classically, only interactive commitments are known under that assumption [50]), but nothing comparable is known for statistically-hiding commitments.

Task 3: Commitments. *Develop statistically-*hiding *collapse-binding commitment schemes under weaker assumptions (e.g., one-way functions) or more efficiently. Develop commitments with weaker assumptions (or more efficient commitments) by using quantum communication. Identify the relationships between collapse-binding commitments and other definitions of computationally binding commitments (e.g., [26, 30]).*

The above three tasks consider elementary cryptographic building blocks. In contrast, the next task explores applications where hashes are used in essential and non-trivial ways, namely block chain protocols and time-stamping protocols. Such protocols are widely studied in classical cryptography, but to the best of our knowledge, no analysis in the post-quantum setting exists. Yet, we expect non-trivial challenges here: Blockchains will need to be resistant against attacks using Grover's algorithm [41]. Time-stamping protocols are essentially commitments with extra requirements, so we expect to encounter the same difficulties here as with post-quantum secure commitments.

Task 4: Blockchains, time-stamping. *Analyze/design complex protocols related to hash functions such as blockchain protocols and time-stamping protocols. In particular, what security notions do the hash functions need to satisfy (is collapsing sufficient?). What security definitions have to be satisfied? Are existing protocols insecure? How to fix them?*

The above tasks were concerned mostly with post-quantum cryptography, i.e., with classical protocols. The following two tasks explore how to use quantum communication (and possibly computation) to construct protocols for tasks that are impossible classically.

Risk	Effect / mitigation strategy
WP1 Task 1: qRHL cannot be formulated in a way that leads to a useful set of reasoning rules (i.e., different contradicting requirements emerge).	Preliminary work [71] indicates that a suitable qRHL can be formulation. If not, we will search for other formalisms/logics to encode game-based proofs (e.g., rewriting in the style of CryptoVerif [14, 27]). Most of the dependent tasks will be adapted to different degrees to use that formalism instead.
WP1 Task 2: Advanced quantum proof techniques cannot be encoded as qRHL rules, nor represented in other ways in qRHL-based proofs.	We will search for alternative proof techniques. If this also fails, it will block the verification of many advanced quantum protocols. Many post-quantum cryptographic schemes can still be analyzed. We will adapt the selection of case studies (WP3) accordingly.
WP1 Tasks 3–5, WP2 Task 3: Linking qRHL to classical pRHL or the development of decision procedures fails, or their implementation in a tool is prohibitively expensive.	This leads to a lack of automation. Proving the case studies will still be possible, although more work.
WP2 Tasks 2: Adapting EasyCrypt to qRHL is not possible due to mismatches in EasyCrypt's architecture.	We have to extend the prototype tool instead. This will incur extra effort, and lead to a less mature tool within the lifetime of the project. The case studies will still be possible.
WP2 Task 4, WP3 Task 5: Encoding qRHL in Isabelle/HOL turns out to be prohibitively expensive and not feasible during the lifetime of this project.	In that case, we have to accept the larger trusted core of the QuEasyCrypt tool. Human error during the proof design is still avoided, but implementation errors in QuEasyCrypt remain possible.
WP3: Performing advanced security proofs in qRHL turns out to be prohibitively expensive and not feasible during the lifetime of this project.	We will study more modest examples (e.g., only classical post-quantum secure protocols that do not use advanced proof techniques).
WP4: We make insufficient advances on quantum cryptographic protocols.	We use protocols developed by other researchers instead for our case studies.

Figure 1: Since our work is of foundational nature, we cannot guaranteed that all research tasks listed above will be successful. Some may turn out to be provably impossible, or some may turn out to require methods too far beyond current knowledge. We give the main **risks** of this proposal and our **mitigation strategies**.

Obfuscation is the process of transforming a program (or circuit) into another program (or circuit) with the same functionality, yet without leaking any information about the inner workings of the program. [8] proved that a general form of obfuscation (virtual black box obfuscation, VBB) is impossible in general. Due to this, cryptographers currently focus more on a much weakened form of obfuscation, called indistinguishability obfuscation (iO) which has candidates [40]. However, the impossibility results from [8] does not apply to the quantum case – it might be possible to obfuscate a classical program into a new representation as a quantum state that satisfies the VBB definition.[11]

Task 5: Quantum obfuscation. *Develop quantum VBB obfuscation schemes, e.g., a construct a quantum VBB scheme from a classical (but quantum secure) iO scheme. Alternatively, prove that quantum VBB schemes are impossible in general.*

No information can be transmitted faster than light. We can base the security of protocols on that fact, this leads to so-called relativistic protocols. For example, position verification protocols [44, 61, 69] allow a device to prove its location in space by relying on quantum communication and the

[11] Note that obfuscation is different from the similar sounding task of blind quantum computation (BQC) [18]. The latter allows interactivity which makes the task much simpler than obfuscation. (BQC is more closely related to multiparty computation than to obfuscation.)

speed of light ([20] shows that without quantum communication, this is not possible). Relativistic commitments use the speed of light to implement information-theoretically commitments [43, 48]. However, it is not known whether practically efficient constructions such as the one from [48] are quantum secure (classical security was shown in [48, 36]). In addition, unexpected subtleties arise in the quantum case: Quantumly, committing to a certain value effectively measures that value. But a relativistic commitment may "expire", and this may retroactively undo this measurement. Security notions for relativistic commitments need to take this into account. (For example oblivious transfer (OT) protocols such as [13] rely on the fact that commitments imply measurements.)

Task 6: Relativistic protocols. *Develop position-verification protocols under relaxed assumptions and/or with higher precision than [69]. ([69] relies on random oracles.) Develop security notions for the quantum security of relativistic commitments (these may also give additional insights for non-relativistic commitments). Develop quantum secure relativistic commitments (or prove known ones secure).*

References

[1] S. Abramsky and B. Coecke. "A categorical semantics of quantum protocols". In: *LICS '04*. IEEE, 2004, pp. 415–425.

[2] R. Adams. "QPEL – Quantum Program and Effect Logic". In: *QPL 2014*. Vol. 172. EPTCS. 2014, pp. 133–153.

[3] *Adecs – A decision algorithm for proving symbolic equivalence of constraint systems*. `https://members.loria.fr/vcheval/tools/adecs/`. Accessed 2016-12-30.

[4] T. Altenkirch and J. Grattage. "A functional quantum programming language". In: *LICS' 05*. June 2005, pp. 249–258.

[5] A. Ambainis, A. Rosmanis, and D. Unruh. "Quantum Attacks on Classical Proof Systems (The Hardness of Quantum Rewinding)". In: *FOCS 2014*. IEEE, 2014, pp. 474–483.

[6] E. Ardeshir-Larijani, S. J. Gay, and R. Nagarajan. "Equivalence Checking of Quantum Protocols". In: *TACAS 2013*. Springer, 2013, pp. 478–492.

[7] E. Ardeshir-Larijani, S. J. Gay, and R. Nagarajan. "Verification of Concurrent Quantum Protocols by Equivalence Checking". In: *TACAS 2014*. Springer, 2014, pp. 500–514.

[8] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang. "On the (Im)Possibility of Obfuscating Programs". In: *J. ACM* 59.2 (2012), 6:1–6:48.

[9] G. Barthe, B. Grégoire, S. Heraud, and S. Zanella Béguelin. "Computer-Aided Security Proofs for the Working Cryptographer". In: *Crypto 2011*. Vol. 6841. LNCS. Springer, 2011, pp. 71–90.

[10] G. Barthe, B. Grégoire, and S. Zanella Béguelin. "Formal Certification of Code-Based Cryptographic Proofs". In: *POPL 2009*. ACM, 2009, pp. 90–101.

[11] M. Bellare and P. Rogaway. "The Security of Triple Encryption and a Framework for Code-based Game-playing Proofs". In: *Eurocrypt 2006*. Vol. 4004. LNCS. Springer, 2006, pp. 409–426.

[12] C. H. Bennett and G. Brassard. "Quantum cryptography: Public-key distribution and coin tossing". In: *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing 1984*. IEEE Computer Society, 1984, pp. 175–179.

[13] C. H. Bennett, G. Brassard, C. Crépeau, and M.-H. Skubiszewska. "Practical Quantum Oblivious Transfer". In: *Crypto '91*. Vol. 576. LNCS. Springer, 1991, pp. 351–366.

[14] B. Blanchet. "A Computationally Sound Mechanized Prover for Security Protocols". In: *SSP 2006*. IEEE, 2006, pp. 140–154.

[15] B. Blanchet, M. Abadi, and C. Fournet. "Automated verification of selected equivalences for security protocols". In: *J Logic and Algebraic Programming* 75.1 (2008), pp. 3–51.

[16] D. Boneh, O. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. "Random oracles in a quantum world". In: *Asiacrypt 2011*. Springer, 2011, pp. 41–69.

[17] N. J. Bouman and S. Fehr. "Sampling in a Quantum Population, and Applications". In: *Crypto 2010*. Vol. 6223. LNCS. Springer, 2010, pp. 724–741.

[18] A. Broadbent, J. Fitzsimons, and E. Kashefi. "Universal Blind Quantum Computation". In: *FOCS 2009*. IEEE, 2009, pp. 517–526.

[19] C. Bădescu and P. Panangaden. "Quantum Alternation: Prospects and Problems". In: *QPL 2015*. Vol. 195. EPTCS. 2015, pp. 33–42.

[20] H. Buhrman, N. Chandran, S. Fehr, R. Gelles, V. Goyal, R. Ostrovsky, and C. Schaffner. "Position-Based Quantum Cryptography: Impossibility and Constructions". In: *Crypto 2011*. Vol. 6841. LNCS. Springer, 2011, pp. 429–446.

[21] *CertiCrypt: Computer-Aided Cryptographic Proofs in Coq*. `http://certicrypt.gforge.inria.fr/`. Accessed 2016-12-29.

[22] R. Chadha, P. Mateus, and A. Sernadas. "Reasoning About Imperative Quantum Programs". In: *Electron. Notes Theor. Comput. Sci.* 158 (May 2006), pp. 19–39.

[23] V. Cheval, H. Comon-Lundh, and S. Delaune. "Automating Security Analysis: Symbolic Equiv-

alence of Constraint Systems". In: *IJCAR 2010*. Springer, 2010, pp. 412–426.

[24] K. Cho, B. Jacobs, B. Westerbaan, and A. Westerbaan. *An Introduction to Effectus Theory*. arXiv:1512.05813 [cs.LO]. 2015.

[25] B. Coecke and A. Kissinger. *Categorical Quantum Mechanics I & II*. arXiv:1510.05468 [quant-ph] and arXiv:1605.08617 [quant-ph]. 2016.

[26] C. Crépeau, P. Dumais, D. Mayers, and L. Salvail. "Computational Collapse of Quantum State with Application to Oblivious Transfer". In: *TCC 2004*. Vol. 2951. LNCS. Springer, 2004, pp. 374–393.

[27] *CryptoVerif: Cryptographic protocol verifier in the computational model*. `http://prosecco.gforge.inria.fr/personal/bblanche/cryptoverif/`. Accessed 2016-12-29.

[28] J. Czajkowski, L. Groot Bruinderink, A. Hülsing, C. Schaffner, and D. Unruh. "Post-quantum security of the sponge construction". In: *PQCrypto 2018*. LNCS. to appear, IACR ePrint 2017/771. Springer, 2018.

[29] I. Damgård, S. Fehr, R. Renner, L. Salvail, and C. Schaffner. "A Tight High-Order Entropic Quantum Uncertainty Relation with Applications". In: *Crypto 2007*. Vol. 4622. LNCS. Springer, 2007, pp. 360–378.

[30] I. Damgård, S. Fehr, and L. Salvail. "Zero-Knowledge Proofs and String Commitments Withstanding Quantum Attacks". In: *Crypto 2004*. Vol. 3152. LNCS. Springer, 2004, pp. 254–272.

[31] E. D'hondt and P. Panangaden. "Quantum Weakest Preconditions". In: *Mathematical. Structures in Comp. Sci.* 16.3 (June 2006), pp. 429–451.

[32] D. Dolev and A. C. Yao. "On the Security of Public Key Protocols". In: *IEEE Transactions on Information Theory* 29.2 (1983), pp. 198–208.

[33] *EasyCrypt: Computer-Aided Cryptographic Proofs*. `https://www.easycrypt.info/trac/`. Accessed 2016-12-29.

[34] *EasyCrypt issues 17132, 17114, 17168*. `https://www.easycrypt.info/trac/ticket/<number>`. Accessed 2017-01-24.

[35] E. Farhi, D. Gosset, A. Hassidim, A. Lutomirski, and P. Shor. "Quantum Money from Knots". In: *ITCS '12*. ACM, 2012, pp. 276–289.

[36] S. Fehr and M. Fillinger. "On the Composition of Two-Prover Commitments, and Applications to Multi-round Relativistic Commitments". In: *Eurocrypt 2016*. Vol. 9666. LNCS. Springer, 2016, pp. 477–496.

[37] Y. Feng, R. Duan, Z. Ji, and M. Ying. "Proof rules for the correctness of quantum programs". In: *Theoretical Computer Science* 386.1 (2007), pp. 151–166.

[38] Y. Feng and M. Ying. "Toward Automatic Verification of Quantum Cryptographic Protocols". In: *CONCUR 2015*. Vol. 42. LIPIcs. Schloss Dagstuhl, 2015, pp. 441–455.

[39] A. Fiat and A. Shamir. "How to Prove Yourself: Practical Solutions to Identification and Signature Problems". In: *Crypto '86*. LNCS 263. Springer, 1987, pp. 186–194.

[40] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. "Candidate Indistinguishability Obfuscation and Functional Encryption for all Circuits". In: *FOCS 2013*. IEEE, 2013, pp. 40–49.

[41] L. K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: *STOC*. 1996, pp. 212–219.

[42] B. Jacobs. "New directions in categorical logic, for classical, proba- bilistic and quantum logic". In: *Logical Methods in Comp. Sci.* 11.3 (2015), pp. 1–76.

[43] A. Kent. "Unconditionally Secure Bit Commitment by Transmitting Measurement Outcomes". In: *PRL* 109.13 (2012), p. 130501.

[44] A. P. Kent, W. J. Munro, T. P. Spiller, and R. G. Beausoleil. *Tagging systems*. US Patent 7,075,438. 2006.

[45] A. Kissinger, S. Tull, and B. Westerbaan. "Picture-perfect QKD". In: *QKD 2017*. EPTCS. to appear, arXiv:1704.08668 [quant-ph]. Open Publishing Association, 2017.

[46] T. Kubota, Y. Kakutani, G. Kato, Y. Kawano, and H. Sakurada. "Application of a Process Calculus to Security Proofs of Quantum Protocols". In: *FCS '12*. 2012, pp. 141–147.

[47] T. Kubota, Y. Kakutani, G. Kato, Y. Kawano, and H. Sakurada. "Automated Verification of Equivalence on Quantum Cryptographic Protocols". In: *SCSS 2013*. Vol. 15. EPiC Series in Computing. EasyChair, 2013, pp. 64–69.

[48] T. Lunghi, J. Kaniewski, F. Bussières, R. Houlmann, M. Tomamichel, S. Wehner, and H. Zbinden. "Practical Relativistic Bit Commitment". In: *Phys. Rev. Lett.* 115 (3 2015), p. 030502.

[49] V. Lyubashevsky. "Lattice-Based Identification Schemes Secure Under Active Attacks". In: *PKC '08*. Vol. 4939. LNCS. Springer, 2008, pp. 162–179.

[50] M. Naor. "Bit Commitment Using Pseudorandomness". In: *J Cryptology* 4.2 (1991), pp. 151–158.

[51] T. Nipkow, L. Paulson, M. Wenzel, G. Klein, F. Haftmann, T. Weber, and J. Hölzl. *Isabelle*. `https://isabelle.in.tum.de/`. Accessed 2017-01-24.

[52] M. Pagani, P. Selinger, and B. Valiron. "Applying Quantitative Semantics to Higher-order Quantum Computing". In: *POPL '14*. ACM, 2014, pp. 647–658.

[53] *ProVerif: Cryptographic protocol verifier in the formal model*. `http://prosecco.gforge.inria.fr/personal/bblanche/proverif/`. Accessed 2016-12-30.

[54] P. W. Shor and J. Preskill. "Simple Proof of Security of the BB84 Quantum Key Distribution Pro-

tocol". In: *Phys. Rev. Lett.* 85.2 (2000), pp. 441–444.

[55] V. Shoup. *Sequences of games: a tool for taming complexity in security proofs*. IACR ePrint Archive, Report 2004/332. 2004.

[56] E. E. Targhi and D. Unruh. "Quantum Security of the Fujisaki-Okamoto and OAEP Transforms". In: *TCC 2016-B*. Vol. 9986. LNCS. Springer, 2016, pp. 192–216.

[57] A. M. Tavala, S. Nazem, and A. A. Babaei-Brojeny. "Verification of Quantum Protocols with a Probabilistic Model-Checker". In: *Electronic Notes in Theoretical Computer Science* 270.1 (2011), pp. 175–182.

[58] *The Coq Proof Assistant*. `https://coq.inria.fr/`. Accessed 2017-01-24.

[59] The PG dev team. *Proof General (A generic Emacs interface for proof assistants)*. `https://proofgeneral.github.io/`. Accessed 2017-01-02.

[60] M. Tomamichel, C. Schaffner, A. Smith, and R. Renner. "Leftover Hashing Against Quantum Side Information". In: *IEEE Trans. Inf. Theor.* 57.8 (2011), pp. 5524–5535.

[61] M. Tomamichel, S. Fehr, J. Kaniewski, and S. Wehner. "One-Sided Device-Independent QKD and Position-Based Cryptography from Monogamy Games". In: *Eurocrypt*. Vol. 7881. LNCS. Springer, 2013, pp. 609–625.

[62] D. Unruh. *CHSH proof for EasyCrypt*. `https://git.io/vAmvg`. Accessed 2017-01-02.

[63] D. Unruh. "Collapse-binding quantum commitments without random oracles". In: *AsiaCrypt 2016*. Vol. 10032. LNCS. Springer, 2016, pp. 166–195.

[64] D. Unruh. "Computationally binding quantum commitments". In: *Eurocrypt 2016*. LNCS. Springer, 2016, pp. 497–527.

[65] D. Unruh. "Everlasting Multi-Party Computation". In: *Crypto 2013*. Vol. 8043. LNCS. Springer, 2013, pp. 380–397.

[66] D. Unruh. *IsaCrypt GitHub repository*. `https://github.com/dominique-unruh/IsaCrypt`. Currently undergoing refactoring!

[67] D. Unruh. "Non-interactive zero-knowledge proofs in the quantum random oracle model". In: *Eurocrypt 2015*. Vol. 9057. 2015, pp. 755–784.

[68] D. Unruh. "Post-quantum Security of Fiat-Shamir". In: *Asiacrypt 2017*. Vol. 10624. Springer, 2017, pp. 65–95.

[69] D. Unruh. "Quantum position verification in the random oracle model". In: *Crypto 2014*. Vol. 8617. LNCS. Springer, 2014, pp. 1–18.

[70] D. Unruh. "Quantum Proofs of Knowledge". In: *Eurocrypt 2012*. Vol. 7237. LNCS. Springer, 2012, pp. 135–152.

[71] D. Unruh. *Quantum Relational Hoare Logic*. arXiv:1802.03188 [quant-ph]. Feb. 2018.

[72] D. Unruh. "Revocable quantum timed-release encryption". In: *Eurocrypt 2014*. Vol. 8441. LNCS. Springer, 2014, pp. 129–146.

[73] J. Watrous. "Zero-Knowledge against Quantum Attacks". In: *SIAM J. Comput.* 39.1 (2009), pp. 25–58.

[74] S. Wiesner. "Conjugate coding". In: *SIGACT News* 15.1 (1983), pp. 78–88.

[75] J. Yan, J. Weng, D. Lin, and Y. Quan. "Quantum Bit Commitment with Application in Quantum Zero-Knowledge Proof". In: *ISAAC 2015*. Springer, 2015, pp. 555–565.

[76] M. Ying. "Floyd–Hoare Logic for Quantum Programs". In: *ACM Trans. Program. Lang. Syst.* 33.6 (Jan. 2012), 19:1–19:49.

[77] M. Ying. *Foundations of Quantum Programming*. 1. edition. Morgan Kaufmann, 2016.

[78] M. Ying, N. Yu, and Y. Feng. *Defining Quantum Control Flow*. arXiv:1209.4379 [quant-ph]. 2012.

[79] M. Zhandry. "Secure Identity-Based Encryption in the Quantum Random Oracle Model". In: *Crypto 2012*. Vol. 7417. LNCS. Springer, 2012, pp. 758–775.