Formal Verification Techniques for Post-Quantum Cryptography: A Systematic Review

Yuexi Xu[§]
University of Queensland
yuexi.xu@uq.net.au

Zhenyuan Li[§]
University of Queensland
zhenyuan.li1@uq.net.au

Naipeng Dong
University of Queensland
n.dong@uq.edu.au

Veronika Kuchta
Florida Atlantic University
vkuchta@fau.edu

Zhe Hou
Griffith University
z.hou@griffith.edu.au

Dongxi Liu
Data61, CSIRO
dongxi.Liu@data61.csiro.au

***Abstract*—In the quantum computing era, the imperative role of post-quantum cryptography in securing digital communications has led to the development of computer-aided cryptography verification tools. These tools simplify the verification of post-quantum cryptography primitives and protocols, alleviating the challenges associated with manual proofs. This paper systematically reviews research in four main areas: quantum computing, post-quantum cryptography, cryptanalysis, and verification, establishing a foundation for future research. Emphasising the significance of challenges in post-quantum cryptography, we outline the current state of research on cryptography primitives and protocols. Categorising state-of-the-art computer-aided cryptography verification tools based on assumptions, models, and application levels, our analysis delves into each tool's features, including modelling, adversary models, security properties, validation, and an in-depth analysis of their limitations. This comprehensive analysis offers insights into the nexus of post-quantum cryptography and computer-aided verification. Concluding with recommendations for researchers and practitioners, this paper explores potential future research directions.**

***Index Terms*—Formal Verification, Post-quantum Cryptography, Review, Survey**

I. Introduction

Cryptography holds significant importance in safeguarding confidential data and communication, serving as the fundamental building block for secure systems. In the rapidly evolving landscape of modern cryptography, the emergence of quantum computing necessitates a comprehensive understanding of their latest advancements and security implications, as the proliferation of quantum computing threatens the security foundations of certain classical cryptographic schemes, calling for an urgent exploration of alternative paradigms.

Simultaneously, there is a pressing need for the evaluation of the robustness of cryptographic primitives and protocols (applications of cryptographic primitives). Thus, cryptanalysis has gained paramount importance, ensuring their correctness, resilience, and alignment with desired security properties. Researchers have turned to formal verification techniques, which provide a systematic and rigorous approach to verifying their reliability and correctness. These techniques play a crucial role in identifying vulnerabilities, errors, and weaknesses that might otherwise go unnoticed.

This survey paper embarks on an ambitious journey through the realms of post-quantum cryptography with breakdowns in four areas: post-quantum cryptography, quantum computing, cryptanalysis, and formal verification, to unravel the intricate tapestry that underpins these domains. Comprehensive research of these areas is necessary because the interplay between them has created a complex and rapidly changing landscape. A deep understanding of the key elements in play is crucial to staying ahead of potential threats and ensuring the robustness of cryptographic applications in this shifting environment. By doing so, this survey provides an overarching perspective that contributes to the ongoing efforts to fortify the digital security infrastructure.

Existing surveys (detailed in Section III) offer valuable insights into specific facets of quantum computing, post-quantum cryptography, cryptanalysis, and verification, but exhibit limitations in their integration—the formal verification of post-quantum cryptography. In addition, they are limited in coverage and depth. Notably, these surveys tend to emphasize theoretical explorations, often overlooking the practical dimensions essential for real-world applications. The challenges and opportunities transitioning the verification approaches from classical to post-quantum cryptography remain largely unexplored, leaving a critical void in our understanding.

To address these gaps, this paper undertakes a comprehensive and meticulous analysis, delving into various verification methods within the context of post-quantum cryptography and protocols. This exploration encompasses a wide range of verification approaches, including manual, automated, and semi-automated methods, while also extending its focus to encompass both classical and quantum paradigms. The survey aims to provide a holistic understanding of the intricacies and potential remedies that define the cryptographic landscape. Through this endeavour, the paper aims to offer insights that will not only illuminate the current panorama but also guide future research directions in the pursuit of robust cryptographic systems capable of withstanding the challenges of an increasingly complex and quantum-powered world.

[§]Yuexi Xu and Zhengyuan Li are co-first authors.

II. Survey Scope

The emergence of **quantum computing** presents a formidable challenge to classical cryptography and its applications, as it can break widely-used cryptography algorithms, such as the RSA signature and the Elliptic Curve Diffie-Hellman (ECDH) based cryptography, by leveraging its superior computational capabilities, specifically for factoring large numbers and solving discrete logarithm problems efficiently. Quantum computing introduces a pressing need for post-quantum cryptography, which aims to develop cryptographic algorithms resilient to quantum attacks, and necessitates a reevaluation of cryptographic systems and their vulnerabilities. Thus, understanding quantum computing is pivotal to grasping the current and future cryptography landscape.

As a countermeasure to the quantum threats, **post-quantum cryptography** (PQC) is proposed that focuses on developing cryptographic schemes and their applications that can withstand quantum attacks. With the rapid improvement of quantum computing, the need for post-quantum cryptographic solutions is urgent [1]. Understanding post-quantum cryptography is also vital to the development and analysis of secure sensitive information and communication systems in the face of emerging quantum threats.

As the security of cryptography is ensured generally by **cryptanalysis** that uncovers vulnerabilities and weaknesses of the cryptography, cryptanalysis of post-quantum cryptography is indispensable for assessing the robustness of the proposed post-quantum cryptographic schemes and the effectiveness of their applications. As such, we will also explore the surveys that evaluate the strengths and weaknesses of both classical and post-quantum cryptographic solutions.

Among various cryptoanalysis, **formal verification** techniques provide a systematic and rigorous approach to verifying the reliability and correctness of cryptographic schemes and applications. It is becoming a norm that a cryptographic scheme must come with a proof that it satisfies some standard security properties to be accepted. These techniques are essential for ensuring that cryptographic systems (including cryptographic schemes and applications) operate as intended and are free from vulnerabilities and errors. Therefore, the study of verification methods, both manual and automated, is crucial to understanding their capabilities, pros and cons in proving correctness and security, contributing to their overall reliability and effectiveness.

This survey explores the above four topics due to their paramount significance and interrelated nature. We examine the interplay between the four topics as they collectively shape the evolving landscape of future digital security. In particular, our focus is on *formal verification techniques for post-quantum cryptography*.

Data Collection. Our investigation encompassed an examination of scholarly works published from 2020 onwards for the above four topics. In the pursuit of a robust comprehension of verification methods, we broadened our scope to encompass literature dating back to 2010, as discerned by the paucity of available information within the field. Employing a methodical keyword-based search strategy, we navigated the scholarly terrain via Google Scholar and databases including ACM, IEEE, ANST and Scopus employing the related keywords, and refined the search results by screening their titles and abstracts.

III. Existing Surveys

There have been surveys on each individual area of quantum computing, post-quantum cryptography, cryptanalysis and formal verification, presented as follows. However, there is no survey focusing on the formal verification of post-quantum cryptography, which is the key motivation of this work.

a) Quantum Computing: Surveys within the domain of quantum computing offer insights into emerging technologies and fields such as machine learning [2], [3], post-quantum cryptography and quantum entanglement [4], and financial field [5] e.g., blockchain [1]. Each paper undertakes a comprehensive review of recent developments and research within its field and provides a discussion of potential areas for future research. Nonetheless, these papers have insufficient coverage of practical implementation challenges, scalability issues, and ethical considerations. Furthermore, these papers tend to prioritize the theoretical physics aspects of quantum computing over other aspects.

b) Post-Quantum Cryptography: Four survey papers are examined on this topic, covering different aspects. The work of Tan et al. [6] focuses on the challenges of implementing post-quantum digital signature algorithms in real-world applications, while the work by Kumari [7] covers the suitability of post-quantum cryptography techniques for securing communication in resource-constrained IoT devices. The work conducted by Hasija and other researchers [8] covers the third-round candidates for post-quantum cryptography selected by NIST and their algorithmic structures, security properties, and implementation details. Finally, the paper of Zeydan et al.'s [9] covers the recent advances in post-quantum cryptography for network security, including key exchange, signature schemes, and encryption schemes. However, none of the papers discuss the challenges and considerations of transitioning from classical to post-quantum cryptography, such as the compatibility of post-quantum algorithms with existing systems or the impact of quantum computers on current cryptographic protocols.

c) Cryptanalysis: There are several papers on cryptography that cover various topics such as encryption methods, key management, digital signatures, and hash functions. Research conducted by Mnkash [10] discusses traditional and advanced encryption techniques, cryptographic protocols, and key distribution methods. Abinaya and Prabakeran [11] focus on lightweight block ciphers and their suitability for IoT devices, covering their design principles, security features, and potential attacks. Wang's work [12] surveys the use of lattice-based cryptosystems in the standardization processes of cryptographic algorithms, highlighting the security properties, potential advantages, ongoing efforts, and open problems in this area. The last paper is a bibliometric analysis of research papers related to the cryptanalysis of block ciphers in cyber

security [13], covering the evolution of block ciphers and cryptanalysis techniques, their application in cyber security, and research trends and patterns. However, these surveys are also theory-focused but light on practical concerns.

d) Formal Verification: Previous survey papers on cryptographic primitive and protocol verification have primarily focused on the blockchain field [14], while others have concentrated on specific areas such as IoT devices [7], [15], algorithms [16] and networks [17]. Matteo et al. [18] conducted a comprehensive analysis of various automated protocol provers used for classical cryptography. Their analysis evaluated RCF, Applied pi-calculus, CryptoVerif, ASPIER, Horn clauses, First-order logic (FOL), and LySa with respect to both computational and symbolic models. The study elaborated on the operational patterns, languages, and formal semantics of each model. In addition to protocol provers, their survey paper also covered code generation approaches. Some survey papers toughed a bit on post-quantum cryptography; for instance, Kumari et al. [7] covered post-quantum cryptographic techniques but specific to the field of IoT devices.

One can observe that there is limited research on formal verification for post-quantum cryptography. Therefore, this paper endeavours to bridge the gap by conducting an extensive survey of verification approaches in the context of post-quantum cryptographic primitives and protocols. Through this exploration, the paper seeks to shed light on the multifaceted challenges, solutions, and potential avenues for future research in the realm of post-quantum cryptographic verification.

Compared with the above formal verification surveys, our work takes a more extensive approach and explores the complete landscape of provers for both primitives and protocols. The scope of our survey is not limited to automated provers but also considers semi-auto and manual provers. Most importantly, we focus on the quantum adversary in our analysis, rather than the polynomial adversary in the classical setting. In contrast to the verification of a specific cryptographic scheme, our paper emphasizes the generic approaches.

IV. Post-quantum Cryptography

Cryptographic primitives and cryptographic protocols are two fundamental components in the field of cryptography. Cryptographic protocols, built on top of primitives, provide rules guiding secure communication between entities. In essence, protocols are applications of primitives. We discuss their state-of-the-art separately in the following subsections.

A. Post-quantum Cryptographic Primitives

Post-quantum cryptographic primitives are often categorized into five classes based on hard problems: Hash-based, Code-based, Multivariate, Lattice-based and Isogenies [1]. Many have been proposed but later found flawed e..g, in the process of NIST competition, highlighting the importance of cryptanalysis. Hence, we only consider the candidates selected by NIST because they have a high chance of being adopted in the real-world applications.

The National Institute of Standards and Technology (NIST) initiated the PQC Standardization project to identify and promote cryptographic algorithms that can defend the evolving threat in the face of quantum computing advancements. After six years of cryptanalysis, in 2022, NIST made a significant stride in this endeavour by officially unveiling a set of cryptographic algorithms deemed as the "selected candidates" in round three. There are four notable candidates [19]: CRYSTALS-KYBER, CRYSTALS-DILITHIUM, FALCON, and SPHINCS+. They fall into two major categories: public-key encryption and key-establishment, and the digital signature. CRYSTALS-KYBER is the exclusive algorithm within the category of public-key encryption and key-establishment. Notably, while the other candidates are rooted in lattice-based cryptographic principles, SPHINCS+ stands out as a hash-based cryptographic solution.

NIST has already initiated the process of collecting feedback and comments for the upcoming fourth round [20], aiming at more public-key encryption and key-establishment algorithms. The submissions in this round encompass BIKE, Classic McEliece, HQC, and SIKE. Note that SIKE employs isogeny-based cryptographic techniques, while the other candidates are code-based.

B. Post-quantum Cryptographic Protocols

In the field of post-quantum cryptographic protocols, current research results are mainly focused on TLS and blockchain, illustrating the limited practical applications of post-quantum cryptography in solving real-world quantum threats.

The Transport Layer Security (TLS) protocol, especially version 1.3, is the foundation for securing data transmission over the internet [21], [22]. Amidst the evolution of quantum computing, TLS 1.3 needs to be validated and improved against quantum threats. Post-quantum TLS candidates are proposed by integrating post-quantum cryptographic primitives [21], [22]. These works also conducted performance experiments, providing a well-reasoned reference for future research and practical preparations in the quantum era.

Blockchain technology has become a dynamic focus in post-quantum protocol studies, evident in numerous papers and studies [1], [23], [24]. The growing volume of research and applications underscores the industry's recognition of blockchain's pivotal role in fortifying the foundations against quantum uncertainties. Researchers are enhancing cryptographic primitives to secure blockchain transactions, focusing on various signature schemes. The convergence of blockchain and post-quantum cryptography addresses the urgent security needs of digital currencies, reshaping trust structures for decentralized systems in a quantum future.

V. Formal Verification Techniques: Taxonomy

Formal verification is a promising way to verify the security of post-quantum cryptographic primitive and protocols. As a rigorous approach, formal verification needs to clearly define the *assumptions* of the results. Given the assumption, and a rigorous *model* representing the system behaviour as well

as the security claims, formal verification applies various *verification techniques* to get the results—whether the model satisfies the security property under the given assumption, with the assistance of computers. We classify the existing formal verification methods for post-quantum cryptography based on the above steps.

The initial classification pertains to the foundational assumptions, differentiating between symbolic and computational approaches. Directly verifying a cryptography-involved system using computers are complex. This is true even in the proving of a cryptographic primitive, e.g., encryption and signature schemes, not mentioning protocols that use these cryptographic primitives. Therefore, abstraction techniques emerge targeting verifying cryptographic protocols. Starting from the Dolev-Yao model [25], a line of research has been developed and matured on automated verification of logic flaws in cryptographic protocols. Notable tools include ProVerif [26] and Tamarin [27]. This research made the following assumptions: 1) The data in a cryptographic protocol are atomic symbols. 2) The cryptographic primitives are assumed to be perfect. 3) The attacker controls the entire network. This approach is limited to protocols only, as verifying cryptographic primitives requires to represent the data in its original form as bitstring (thus, cryptographic primitives are assumed to be perfect). And it limits the verification capability to logic flaws only.

Due to the high level of abstraction, the symbolic approach is far from the view usually adopted by cryptographers. On the contrary, the computational approach treats data as bitstrings and, therefore, is able to represent cryptographic primitives and be able to prove a different type of security properties represented as games (see Section VI). Note that we refrain from a detailed examination of tools based on symbolic assumptions and models, such as ProVerif and Tamarin, as these have been comprehensively surveyed and studied [28]. Furthermore, this classification predominantly focuses on classical cryptography, which does not align with the core objectives of our research.

Under the computational assumption, the second classification pertains to the underlying models that describe the behaviour of the cryptography. The first class inherits the symbolic models but is able to prove computational properties. Notable approaches are PQ-Squirrel [29], based on the work Squirrel [30]. The other class represents the behaviour in computational models.

Subsequently, the computational model category has two categories depending on their verification approaches: CryptoVerif [31] and the EasyPQC [32]. EasyPQC is based on previous works CertiCrypt [33] and EasyCrypt [34].

Finally, from the application perspective, we distinguish the approaches into two categories: one focuses on verifying cryptographic primitives; the other on the protocols, as they adopt different adversary and abstraction levels. Before detailing the verification techniques, we introduce some basic concepts.

VI. Game-based Cryptography Verification

Proving security of cryptographic primitives can be tedious work that tends to be error-prone and difficult to read, due to the non-trivial mathematics that are involved such as number theory, group theory and probability theory. To reduce the proof complexity, a game-based approach is proposed in 2004 [35]. The game is played between an *adversary* and some benign entity called the *challenger*, where the adversary and challenger are probabilistic processes that communicate with each other. Security, in this setting, is defined as some particular events S occurring (e.g., adversary guessing a bit) is bounded by a target probability that is normally very small [35], meaning that the adversary does not gain any advantage in guessing the secret message.

The proof follows a refinement approach of games, sketched as follows: Security of cryptography is based on the hard maths problems that are difficult to solve. If a problem is hard enough, it would take too long for an attacker with even quantum computers to solve it. Therefore, proving the security of a cryptographic algorithm boils down to proving its equivalence to the underlying hard maths problems. Therefore, to prove the security property, the game-based approach creates a sequence of games, denoted as G_0, G_1, ..., G_n. The G_0 is the original game describing the target cryptographic primitive that needs to be proved. In the sequence, a successive game only modifies a small detail or a little step of the previous game. Normally there are three types of transitions between the successive games: 1) transitions based on indistinguishability where the changes are indistinguishable by an efficient distinguisher (the adversary); 2) transitions based on failure event where the two games are identically unless a certain "failure event" occurs and thus probability is bounded by the probability of the "failure event"; and 3) bridging steps which aim to prepare the ground for a transition of one of the above two types.

This refinement of the game-based approach is less error-prone and more easily verifiable, even mechanically verifiable [36]. Following the direction of mechanical verifiability, Bellare and Rogaway deemed games and adversaries as programs and proposed a programming language to represent a game [37]. Halevi took a step further and proposed the idea of creating a computer-aided tool for generating proofs to reduce human errors and increase proof readability [38]. The basic idea is that a game is automatically generated (by the tool) that consists of a main loop where each iteration calls an adversary routine, supplying it with the results of the last iteration and getting back the results of the current iteration. The output is typically the adversary output of the last iteration. To prove the security of the game, the tool generates a sequence of the above games, each time changing some aspects of the current game where the changing is within a list of permissible transformations. The proof proceeds until they are reduced to the empty game, where nothing is left in the code to analyze.

Based on the above idea, David proposed a refinement framework for generating proofs using the proof assistant Coq. Corin and Hartog extended the probabilistic Hoare Logic with functions to represent attackers with arbitrary behaviour and orthogonality that allows them to reason about the game transformation [39]. Courant et al. [40] proposed an automated procedure based on Hoare Logic, dedicated to analyzing

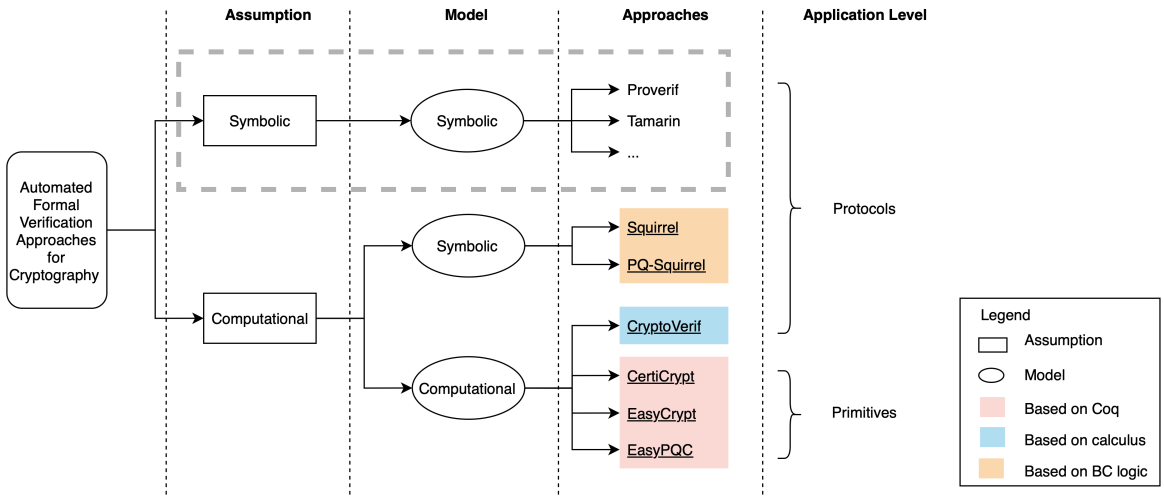

Fig. 1. Taxonomy of Formal Verification Techniques for Post-Quantum Cryptography

asymmetric encryption schemes. Later, these methods were adopted in [41], [42] to verify symmetric encryption modes and message authentication codes. Based on these game-based approaches[§], formal verification of post-quantum cryptography is developed, detailed in Section VII.

VII. Verification of Cryptographic Primitives

In this section, we discuss the verification techniques focusing on proving post-quantum cryptographic primitives—EasyPQC. To do so, we first introduce the two prior works—EasyCrypt and CertiCrypt, on which EasyPQC is based.

A. CertiCrypt

Following the game-based approach, CertiCrypt is proposed, aiming to build an automated framework designed to facilitate the verification of (classical) cryptography utilizing the Coq proof assistant. Compared to other works, CertiCrypt has improved support for proof automation and wide applications e.g., being able to handle random oracles, security assumptions such as Diffie–Hellman hardness assumption, and probabilistic polynomial time complexity [33]. It adopts the code-based approach, meaning that the security goals and hardness problems are modelled as probabilistic programs with unspecified adversary code, uses tools issued from program verification and programming language theory to rigorously check cryptographic reasoning.

To verify a cryptographic primitive, the analysts need to rigorously specify the following components: a model representing the cryptography algorithm, an adversary model, and the security property. Given these information, CertiCrypt developed verification techniques to automate the proofs. We detail the above components as follows:

1) Modeling: To specify the game/code, CertiCrypt proposed an imperative programming language pWHILE with probabilistic assignments, structured datatypes, and procedure calls [33]. It is constructed upon the WHILE-programming language, structured to encompass functionalities of assignments, if-then-else statements, and while loops [45]. In addition, CertiCrypt ensures well-typed expressions and commands and allows user-defined types, operators, as well as general types, including booleans, bitstrings, natural numbers, pairs, lists, and elements in a group. For the game formalization, CertiCrypt additionally offers a definition that maps a procedure in games to commands that are consistent with its parameters, body and return expression.

2) Adversary Model: CertiCrypt assumes a common adversary whose computational complexity is bounded by a polynomial function. This is achieved by specifying variables and procedures that are accessible to adversaries. The adversary can call oracles, but any other procedures it calls must follow a given set of rules that guarantee that each time the adversary reads and writes a variable, the adversary has the required permission. Additional constraints may be imposed on adversaries and are formalized using lists that record the oracle calls and verify that the calls are legitimate.

3) Security Property: The main difference between CertiCrypt and other tools is that CertiCrypt provides exact security rather than showing asymptotically negligible advantages for any effective adversary against the security of a cryptographic system [33]. The exact security is to provide a concrete lower and upper bound for the advantage of an adversary execution, where the lower bound is the probability that the adversary successfully breaks the hard problem and the upper bound is the execution time [33].

[§]Formal verification approaches exist that are not game based, for instance, a type system that tracks whether values are uniform and fresh, or adversarial controlled, is proposed and used for classical cryptographic primitive verification [43], [44].

4) Verification: CertiCrypt uses a code-based technique that relies on programming theory to justify the process of proof. It offers concrete tools to reason about the equivalence of probabilistic programs involving relational Hoare logic, observational equivalence theory, reasoning based on sequences of games technique, and verified program transformation [33].

CertiCrypt develops a Probabilistic Relational Hoare Logic (pRHL) to reason about the equivalence of programs, by extending the relational Hoare logic (RHL) [33]. Both RHL and pRHL are extensions of the classical Hoare logic; the main difference is that RHL is used for deterministic programs while pRHL is used for probabilistic programs. In detail, in RHL, a program fragment starts in a state satisfying the precondition and will terminate in a state satisfying the postcondition; while in pRHL, the logic additionally deals with assertions about probabilistic properties, for example, the probability of a certain event happening with a certain value. The reasoning of pRHL relies on *judgments* in the form of $c_1 \sim c_2 : \Phi \Rightarrow \Psi$, where c_1 and c_2 are probabilistic programs, and both Φ and Ψ are first-order relational assertions where Φ is the pre-condition and Ψ is the post-condition [34], [46].

In this context, games G_1 and G_2 are equivalent w.r.t. pre-condition Ψ and post-condition Φ iff for any initial memories m_1 and m_2 satisfying the pre-condition $m_1 \Psi m_2$, if the evaluations of G_1 in m_1 and G_2 in m_2 terminate with final memories m_1' and m_2' respectively, and $m_1' \Phi m_2'$ holds, which is formalised as the following judgment:

$$\begin{aligned} \models \quad & G_1 \sim G_2 : \Psi \Rightarrow \Phi \quad \stackrel{\text{def}}{=} \\ & \forall m_1 m_2.\ \ m_1 \Psi m_2 \Rightarrow (G_1) m_1 \sim_\Phi (G_2) m_2. \end{aligned}$$

CertiCrypt developed a set of derived rules to reason about whether the above equivalence can be achieved. It formalised a theory of observational equivalence which is an instance of the above judgment where pre and post-conditions are limited to relations based on equality over a subset of variables. All the derived rules can be specialized to the case of observational equivalence. CertiCrypt implements a calculus of variable dependencies and two functions, such that given a command and a set of output (input) variables, it computes a set of input (output) variables such that the two games are observational equivalent. In addition, CertiCrypt provides a set of tactics and algebraic equivalences to automate the bridging steps.

5) Summary: CertiCrypt is a fully formalized framework dedicated to cryptographic primitives, including OAEP [47], FDH [48], and zero-knowledge protocols [49]. Its limitations are twofold: it can only be used for classical cryptography; creating machine-checked proofs is time-consuming and demands a significant level of proficiency in formal proof.

B. EasyCrypt

EasyCrypt is a dedicated tool for cryptographic primitives, aiming at reducing the effort and expertise required for adopting formal verification techniques by providing enhanced automation, addressing the limitation of CertiCrypt that is not user-friendly. Instead of generating high-guarantee proofs, EasyCrypt builds machine-checked proofs from proof sketches and offers a machine-processable representation of the essence of a security proof [50].

EasyCrypt is built based on the same principle as CertiCrypt; thus, they share the same adversary model and support the same security properties. Hence we omit these two components and focus on the modeling and verification.

1) Modeling: Compared to CetiCrypt, EasyCrypt uses functional language that mainly has two types of declarations: basic declarations and game declarations [34]. EasyCrypt can declare types, constants, and operators as its basic declarations, for example, some basic types: unit, bool, int, real, bitstring, list, and finite map. The game declarations are used to model the games during the security proof involving probabilistic statements such as while, if, function definition with keyword *fun*, adversary declaration with keyword *adversary*, and game definition with keyword *game* [34]. Typically, a game is defined as a module (module) with some procedures (proc) in EasyCrypt. In addition, EasyCrypt inherits some language properties from Coq, for example, it can use tactics.

2) Verification: Similar to CertiCrypt, EasyCrypt supports interactive construction in the form of games where the theoretical foundation for game transitions is probabilistic Relational Hoare Logic (pRHL) and pRHL judgments. Unlike CertiCrypt, EasyCrypt separates the program verification and information-theoretic—while the pRHL is used for logical relations connection between games, pRHL judgements are applied for information-theoretic reasoning of the events [50].

EasyCrypt is more effective compared to CertiCrypt, in that EasyCrypt adopts first-order logic to represent the verification conditions—sufficient and valid conditions for a pRHL judgement, and implements an automated procedure that computes the verification conditions. Since the logic in EasyCrypt does not involve probability, general-purpose theorem provers, for example, the SMT (Satisfiability Modulo Theories) solver, can easily be adopted for generating the proofs. In contrast, CertiCrypt sometimes needs probability-involving proofs and thus is less efficient than EasyCrypt.

Compared to CertiCrypt, EasyCrypt is more user-friendly. Since EasyCrypt is set up based on CertiCrypt, and the verifiable proof sketches are compiled into the CertiCrypt framework and then automatically checked by provers such as SMT solver [50], EasyCrypt inherits the modelling flexibility of CertiCrypt, but the adoption of the general-purpose assistants improves the readability of the proofs [50].

3) Summary: EasyCrypt supports various cryptographic primitives including public-key encryption schemes, block cipher modes of operation, digital signature schemes, and hash function designs [34]. However, most of them are limited to cryptographic primitives. It is not intuitive to implement cryptographic protocols using the EasyCrypt because cryptographic protocols are more complex and consider different adversary. Proving the security of a protocol involves not only proving the security of individual primitives but also considering their interactions and the composition of these primitives. And analyzing these interactions may require a higher level of abstraction and modeling.

C. EasyPQC

EasyPQC is an extension of the EasyCrypt that supports post-quantum cryptography proofs [32]. The main challenge is the ability to reason about quantum adversary that can simultaneously query the quantum random oracle with the constraint that queries cannot be retried [32]. EasyPQC addresses the challenge by extending relational logic from pRHL to qpRHL. The theoretical foundation that empowers EasyPQC is the QROM [51].

1) QROM: A random-oracle model (ROM) plays an important role in the analysis of cryptographic primitives, providing random values in response to queries of an adversary. A ROM is a 'black box' that hides secret information from the adversary, simulating functions that the adversary can call but knows nothing of the internal information. For example, an adversary can query a hash of a message from ROM without knowing the original message. The QROM is a quantum ROM that represents the adversary that is able to call quantum programs/functions. Differing from classical programs, quantum programs allow multiple states (represented as quantum bits) to exist simultaneously. Thus, for a given query, QROM provides the distribution of the final simultaneous multiple states of the quantum program, instead of a single state.

2) Modeling: The EasyPQC defines quantum procedure calls enabling the adversary query the QROM, indicated by the keyword *quantum*. In addition, EasyPQC allows defining quantum variables by providing the commands of quantum initialization (assigning a classical value to quantum variables), unitary transformation (quantum operations) and quantum measurement (assigning quantum values to classical variables).

3) Verification: EasyPQC proposes a moderate variant of EasyCrypt by modifying the theoretical game transition techniques from pRHL to post-quantum relational Hoare logic (pqRHL). It proves that pqRHL is sound for reasoning about quantum adversaries; and in consequence, EasyPQC is sound for post-quantum cryptography (PQC) security proofs [32].

The key technique is that the pqRHL enables probabilistic reasoning of quantum adversaries. To this end, pqRHL first extends the relational equality assertions by defining a global equality operation to represent the quantum assertions. The classical assertions cannot be applied because the state of a quantum adversary is non-deterministic and multiple states exist simultaneously [32]. Subsequently, EasyPQC defines two extra rules for reasoning of adversary that incorporates the quantum assertions, and defines rules to support reasoning of the above quantum commands. Other than the above, the rest of the rules are the same as in pRHL; therefore, EasyPQC can adopt the existing proof system of pRHL.

Note that in the proving process, EasyPQC faces another challenge due to the quantum setting. The execution path of the probabilistic/quantum programs is non-deterministic. Consequently, the final output state after a conditional operation (if-then-else) is a mixture of every possible output from each branch. In other words, the program doesn't produce a single deterministic output but rather a mixture of possible outcomes. To ensure that the mixture of outputs also satisfies the preconditions, a set of side conditions named the CM conditions is needed. In the probabilistic setting (classical cryptography), the CM can be checked for each relational proof, however, this is impossible in the quantum setting due to the simultaneously existing states. To address this challenge, EasyPQC develops a theory proving that it is sufficient to additionally checking the satisfaction of CM at the final post-condition. This enables the reuse of the existing proof systems of pRHL by exempting the CM checking in each relational proof and only checking CM for the final post-condition.

4) Summary: Evidenced by the full domain hash signature example, EasyPQC can be used to prove the security of post-quantum cryptography against quantum adversaries [51]. However, only three cryptographic schemes are verified by lifting the classical setting to post-quantum setting, they are PRF-based MAC, Full Domain Hash and GPV08 identity-based encryption [32]. There is no guarantee that other cryptography can be verified in a post-quantum setting. In addition, there is no guarantee that EasyPQC can achieve the formal verification of cryptographic protocols since all EasyPQC's case studies are limited to cryptographic primitives [29].

VIII. Verification of Cryptographic Protocols

This section introduces the verification techniques for post-quantum cryptographic protocols—PQ-Squirrel. Since it is based on the BC logic and is an extension of the Squirrel verifier, we introduce the BC logic and Squirrel first.

A. BC Logic

Bana-Comon (BC) logic is a formal system designed to analyze and reason about (classical) cryptographic protocols [52]. Differing from the mainstream protocol verification techniques that make symbolic assumptions, such as ProVerif and Tamarin, BC logic is built upon a computational model. Also, BC logic focuses on capturing the interactions and computations of the protocol participants as well as the adversary.

1) Modeling: BC logic provides a way to express properties and assertions about the behaviour and security of protocols. It extends the standard first-order logic with additional constructs and operators specific to protocol analysis.

2) Adversary Model: BC logic introduces the notion of a "symbolic attacker" who is capable of performing symbolic computations and reasoning. The attacker is Turing-complete. This enables the analysis of cryptographic protocols under a wide range of computational scenarios and provides a stronger notion of security than previous models.

3) Verification: BC logic incorporates concepts from computational complexity theory, such as polynomial-time reductions and NP-complete problems, to reason about the computational aspects of protocol security. It provides a framework to formally specify equivalence properties of protocols, such as strong equivalence and trace equivalence. Strong equivalence relates to the indistinguishability of protocol executions from an attacker's perspective, while trace equivalence considers the equivalence of protocol traces under various attacks.

B. Squirrel

Squirrel [53] is an interactive and semi-automatic prover whose primary goal is to assist in the formal verification of (classical) cryptographic protocols. Built upon the BC logic, it adopts a symbolic model with a computational security guarantee, which allows for reasoning about the protocol's behaviour in a step-by-step manner.

1) Modeling: Squirrel is based on the pi-calculus and first-order logic. The symbolic Squirrel model captures the behaviour of the protocol by considering the various messages exchanged between the participants and their computational capabilities.

2) Adversary Model: Squirrel takes into account the presence of an adversary who can manipulate the messages and try to break the security properties of the protocol.

3) Verification: The prover employs an interactive approach, where the user and the prover engage in a dialogue to verify the protocol. The user provides the prover with high-level properties that the protocol should satisfy, typically expressed in a formal logic or specification language. The prover then employs various developed techniques and algorithms to reason about the protocol's execution and attempts to prove or disprove the given properties.

4) Summary: Squirrel provides a rich set of built-in cryptographic constructs and primitives that can be used in the protocol specification, and supports verification of various cryptographic primitives and protocols, including encryption, key exchange, digital signatures, secure multi-party computation, and signed DDH protocol.

C. PQ-Squirrel

Classical proof techniques in Squirrel do not carry over to quantum case, as they are incapable of addressing quantum adversaries, because quantum information cannot be copied and measurements destroy information [29]. PQ-Squirrel extends the Squirrel tool with features that simplify protocol specification and verification under post-quantum security assumptions.

1) Modeling: It simplifies the process of specifying protocols by automatically generating attacker terms from input and output commands, assuming the existence of a single attacker. This design choice aligns well with post-quantum requirements and prevents users from inadvertently modelling a weaker threat model with multiple disjoint attackers.

2) Verification: PQ-Squirrel offers two verification modes: classic mode and post-quantum mode. In the post-quantum mode, PQ-Squirrel restricts the use of tactics and axioms to those that have been proven to be post-quantum sound. This is achieved by performing synchronization checks for every indistinguishability appearing in a proof, ensuring that the specified side conditions are met. The main idea is to identify a minimal set of syntactic conditions, resulting in a concise extension comprising only a few hundred lines of additional code. The verification in PQ-Squirrel depends on the PQ-BC logic, which is an extension of the BC logic, as detailed below.

PQ-BC Logic. It was observed that the original proofs in BC can directly apply to the post-quantum setting if there exists an instantiation of the assumption that satisfies the requirement against a quantum attacker. Based on this insight, PQ-BC extends the BC logic by considering the post-quantum soundness of BC rules with respect to cryptographic assumptions.

The cryptographic assumptions supported in PQ-BC include PRF (Pseudorandom Function), IND-CCA (Indistinguishability under Chosen Ciphertext Attack), EUF-CMA (Existential Unforgeability under Chosen Message Attack), ENC-KP (Encryption Key Privacy), INT-CTXT (Integrity of Ciphertext), and OTP (One-Time Pad). To ensure post-quantum security, these assumptions must be instantiated in a manner that is secure against post-quantum attackers. However, the instantiation of the DDH (Decisional Diffie-Hellman) assumption, which is known to be secure against classical attackers, currently lacks a post-quantum secure instantiation and is therefore excluded from the list of allowed cryptographic assumptions in PQ-BC.

IX. Potential Alternatives

In addition to the above dedicated approaches for post-quantum cryptography, we observed an opportunity that proofs in the classical setting may be "lifted" as the proofs in the quantum setting. The researchers in [54] proposed a general framework where quantum security proofs are decomposed into a series of classical security reductions. The study defines the sufficient conditions under which classical reductions can be "lifted" into a quantum environment, including the equivalence of games, the preservation of reduction properties, the structure of linear reductions, and assumptions about particular classes of machines must be met. For instance, the case studies of EasyPQC satisfy these conditions, thus being verified straightforwardly in EasyPQC. Once proved satisfying these sufficient conditions, the various formal verification approaches for classical cryptographic primitives discussed previously can be applied to prove post-quantum cryptography.

It also provides opportunities for proving cryptographic protocols. Since CryptoVerif applies to classical protocol proofs, proving that protocols satisfy the above conditions allows them to be "lifted" to the post-quantum environment by re-using the classical CryptoVerif proofs.

A. CryptoVerif

1) Modeling: The games in CryptoVerif are presented in a process calculus [31], which is developed based on the pi calculus. Messages are represented as bitstrings, and cryptographic primitives are functions that transform bitstrings into other bitstrings. It extends the pi-calculus with a probabilistic choice operator to support the modelling of random variables and defines probabilistic semantics. Indistinguishability is the primary method used to establish security features.

2) Verification: CryptoVerif makes use of a collection of game transformations.

- **Game Initialization**: Cryptoverif typically begins by defining an initial game, often denoted as G_0, which represents the starting point of the protocol analysis. This may involve setting up the protocol's initial state and defining the security properties to be verified.

- **Game Execution** The protocol's progress is modelled as transitions from one game state to another, symbolized by $\rightarrow$. These transitions capture protocol actions and adversarial moves, demonstrating the evolving nature of the analysis: $G_i \xrightarrow{\text{protocol action/adversarial move}} G_{i+1}$.
- **Game Termination Condition** CryptoVerif defined conditions for game termination e.g., restricting sessions.

An essential kind of transformation takes use of the security assumptions made on cryptographic primitives via the use of observational equivalence. Other game transformations are called syntactic transformations, which are used 1) to simplify the game that is acquired after applying an assumption to a cryptographic primitive or 2) to enable the application of an assumption on a cryptographic primitive. These presumptions are defined in the CryptoVerif library. Users are also permitted to change the libraries in order to incorporate cryptographic primitives that are absent from the default library.

These game transformations are structured using a proof approach: when a transformation fails, it proposes alternative transformations that should be applied in order to allow the transformation that is wanted - and this led to the automatic proof generation and automatic game generation of CryptoVerif. A key difference is that CryptoVerif is able to automatically reason about protocols that involve the generation and distribution of random values, such as public key protocols that use randomized public keys or random session keys. This enables CryptoVerif to reason about the security of protocols that rely on randomness in a rigorous and automated way.

B. Specific Approach for Lattice-based Cryptography

In contrast to the above approaches that work for general cryptography, there is a notable work that proposed a symbolic logic to prove the correctness specifically for lattice-based cryptography [55]. Although not directly applicable to post-quantum cryptography, it provides a foundation that could be extended for other lattice-based cryptography for instance the three NIST post-quantum standards. This work is built based on a logic called Computational Indistinguishability Logic (CIL), which is proposed for reasoning about cryptographic primitives in computational models [56]. Game-based proofs are achieved by transforming an oracle system (modelling the initial game) to another bisimilar system until failure. Other works based on CIL exist for proving cryptographic constructions/protocols; but since they are not particularly for post-quantum cryptography, we do not detail them here.

X. Gaps, Limitations and Opportunities

A. Gaps and Limitation of Existing Works

In the process of the survey, we identified the following gaps and limitations of existing works with respect to the post-quantum applications, verification case studies, verification capabilities and tool usability.

a) Application Gaps: We have noticed that there is a gap between theories and applications. While many post-quantum protocols have been proposed in academia, the practical application and verification of these protocols remain limited. The paucity of concrete examples hinders the ability to establish and demonstrate the security properties required to prove the effectiveness of the protocols.

b) Case Study Gaps: Among all the examples provided by various tools, there are not enough up-to-date examples in the NIST final list. In addition, existing theoretical frameworks (e.g., lifting theorems) lack automated application methods and there are gaps in the practical application of these theories.

c) Verification Capability Limitations: The previous sections have illustrated that formal verification of post-quantum cryptography is still in its infancy. There is no theoretical proof of the verification capabilities of existing approaches, nor sufficient case studies to draw a line in the sand. There are limited real-world post-quantum cryptographic applications that are rigorously tested and verified against the desired security properties.

d) Usability: Stemming from the tools' intricate design and complex configuration, they are difficult for researchers to navigate. In addition, limited customization options and platform support limit the adaptability of these tools to different research needs and technical environments. The different program languages used by the tools can also make them less efficient for researchers unfamiliar with them. For instance, EasyCrypt uses Coq and CryptoVerif uses applied pi-calculus; they both have a considerable learning curve.

B. Opportunities

Based on our observation and analysis, the following directions are promising to enhance the post-quantum cryptography and its verification.

- As stated in Section III, existing surveys are insufficient in the related topics of quantum computing, post-quantum cryptography and cryptanalysis of post-quantum cryptography. This survey only addresses the gaps in formal verification of post-quantum cryptography. Comprehensive surveys on the remaining topics help accelerate the adoption of more secure solutions withstanding in the quantum are.
- Introducing and enhancing cryptographic tools by providing researchers and practitioners with more robust resources for analysis and implementation, including making them user-friendly by 1) developing fully automated verification algorithms and providing better tutorials; 2) clarifying their capabilities theoretically or in practice by introducing more case studies.
- As stated in the previous section if there exists a systematic approach to guide users through the validation of conditions specified in [54], the need for introducing a new tool becomes less imperative, thereby offering a pathway to simplify the entire workflow.
- Encouraging and supporting the real-world applications of post-quantum cryptography, fostering their integration into practical systems to validate their effectiveness beyond theoretical frameworks.

The above directions provide ample opportunities for future work in post-quantum cryptography.

REFERENCES

[1] A.-T. Ciulei, M.-C. Crețu, and E. Simion, "Preparation for post-quantum era: a survey about blockchain schemes from a post-quantum perspective," *Cryptology ePrint Archive*, 2022.

[2] S. B. Ramezani, A. Sommers, H. K. Manchukonda, S. Rahimi, and A. Amirlatifi, "Machine learning algorithms in quantum computing: A survey," in *IJCNN*, 2020, pp. 1–8.

[3] Z. Abohashima, M. Elhosen, E. H. Houssein, and W. M. Mohamed, "Classification with quantum machine learning: A survey," *arXiv preprint arXiv:2006.12270*, 2020.

[4] K. Shannon, E. Towe, and O. K. Tonguz, "On the use of quantum entanglement in secure communications: a survey," *arXiv preprint arXiv:2003.07907*, 2020.

[5] D. Herman, C. Googin, X. Liu, A. Galda, I. Safro, Y. Sun, M. Pistoia, and Y. Alexeev, "A survey of quantum computing for finance," *arXiv preprint arXiv:2201.02773*, 2022.

[6] T. G. Tan, P. Szalachowski, and J. Zhou, "Challenges of post-quantum digital signing in real-world applications: a survey," *International Journal of Information Security*, vol. 21, no. 4, pp. 937–952, 2022.

[7] S. Kumari, M. Singh, R. Singh, and H. Tewari, "Post-quantum cryptography techniques for secure communication in resource-constrained internet of things devices: A comprehensive survey," *Software: Practice and Experience*, vol. 52, no. 10, pp. 2047–2076, 2022.

[8] T. Hasija, K. Ramkumar, A. Kaur, S. Mittal, and B. Singh, "A survey on nist selected third round candidates for post quantum cryptography," in *ICCES*, 2022, pp. 737–743.

[9] E. Zeydan, Y. Turk, B. Aksoy, and S. B. Ozturk, "Recent advances in post-quantum cryptography for networks: A survey," in *MobiSecServ*, 2022, pp. 1–8.

[10] S. H. Mnkash, "Survey of different cryptography methods," *Resmilitaris*, vol. 12, no. 2, pp. 4624–4629, 2022.

[11] M. Abinaya and S. Prabakeran, "Lightweight block cipher for resource constrained iot environment—an survey, performance, cryptanalysis and research challenges," *ICICNIS*, pp. 347–365, 2022.

[12] A. Wang, D. Xiao, and Y. Yu, "Lattice-based cryptosystems in standardisation processes: A survey," *IET Information Security*, 2023.

[13] P. A. Bagane and S. Kotrappa, "Bibliometric survey for cryptanalysis of block ciphers towards cyber security," *Library Philosophy and Practice*, pp. 1–18, 2020.

[14] J. Liu and Z. Liu, "A survey on security verification of blockchain smart contracts," *IEEE Access*, vol. 7, pp. 77 894–77 904, 2019.

[15] K. Hofer-Schmitz and B. Stojanović, "Towards formal verification of iot protocols: A review," *Computer Networks*, vol. 174, p. 107233, 2020.

[16] C. Chareton, S. Bardin, D. Lee, B. Valiron, R. Vilmart, and Z. Xu, "Formal methods for quantum programs: A survey," *arXiv preprint arXiv:2109.06493*, 2021.

[17] K.-A. Shim, "A survey of public-key cryptographic primitives in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 577–601, 2015.

[18] M. Avalle, A. Pironti, and R. Sisto, "Formal verification of security protocol implementations: a survey," *Formal Aspects of Computing*, vol. 26, pp. 99–123, 2014.

[19] NIST, "Selected algorithms 2022 - post-quantum cryptography: Csrc," https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022, retrieved on 20/10/2023.

[20] ——, "Round 4 submissions - post-quantum cryptography: Csrc," https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions, retrieved on 20/10/2023.

[21] M. Sosnowski, F. Wiedner, E. Hauser, L. Steger, D. Schoinianakis, S. Gallenmüller, and G. Carle, "The Performance of Post-Quantum TLS 1.3," in *CoNEXT*, 2023, pp. 19–27.

[22] N. Alnahawi, J. Müller, J. Oupický, and A. Wiesmaier, "SoK: Post-Quantum TLS Handshake," *Cryptology ePrint Archive*, 2023.

[23] N. K. Parida, C. Jatoth, V. D. Reddy, M. M. Hussain, and J. Faizi, "Post-quantum distributed ledger technology: a systematic survey," *Scientific Reports*, vol. 13, no. 1, p. 20729, 2023.

[24] T. M. Fernandez-Carames and P. Fraga-Lamas, "Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks," *IEEE access*, vol. 8, pp. 21 091–21 116, 2020.

[25] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[26] "Proverif," https://bblanche.gitlabpages.inria.fr/proverif/.

[27] "Tamarin prover," https://tamarin-prover.com/.

[28] M. Barbosa, G. Barthe, K. Bhargavan, B. Blanchet, C. Cremers, K. Liao, and B. Parno, "Sok: Computer-aided cryptography," in *S&P*, 2021.

[29] C. Cremers, C. Fontaine, and C. Jacomme, "A logic and an interactive prover for the computational post-quantum security of protocols," in *S&P*, 2022, pp. 125–141.

[30] "The squirrel prover," https://github.com/squirrel-prover/squirrel-prover/.

[31] B. Blanchet, "Cryptoverif: A computationally-sound security protocol verifier," *Tech. Rep.*, 2017.

[32] M. Barbosa, G. Barthe, X. Fan, B. Grégoire, S.-H. Hung, J. Katz, P.-Y. Strub, X. Wu, and L. Zhou, "Easypqc: Verifying post-quantum cryptography," in *CCS*, 2021, pp. 2564–2586.

[33] G. Barthe, B. Grégoire, and S. Zanella Béguelin, "Formal certification of code-based cryptographic proofs," in *POPL*, 2009, pp. 90–101.

[34] G. Barthe, F. Dupressoir, B. Grégoire, C. Kunz, B. Schmidt, and P.-Y. Strub, "Easycrypt: A tutorial," *FOSAD*, pp. 146–166, 2012.

[35] V. Shoup, "Sequences of games: a tool for taming complexity in security proofs," *IACR Cryptol. ePrint Arch.*, p. 332, 2004.

[36] M. Bellare and P. Rogaway, "The security of triple encryption and a framework for code-based game-playing proofs," in *EUROCRYPT'06*, 2006, p. 409–426.

[37] ——, "Code-based game-playing proofs and the security of triple encryption," Cryptology ePrint Archive, Paper 2004/331, 2004.

[38] S. Halevi, "A plausible approach to computer-aided cryptographic proofs," *IACR Cryptol. ePrint Arch.*, vol. 2005, p. 181, 2005.

[39] R. Corin and J. den Hartog, "A probabilistic hoare-style logic for game-based cryptographic proofs (extended version)," Cryptology ePrint Archive, Paper 2005/467, 2005, https://eprint.iacr.org/2005/467.

[40] J. Courant, M. Daubignard, C. Ene, P. Lafourcade, and Y. Lakhnech, "Towards automated proofs for asymmetric encryption schemes in the random oracle model," in *CCS*, 2008, p. 371–380.

[41] M. Gagné, P. Lafourcade, Y. Lakhnech, and R. Safavi-Naini, "Automated security proof for symmetric encryption modes," in *Annual Asian Computing Science Conference*, vol. 5913, 2009, pp. 39–53.

[42] M. Gagné, P. Lafourcade, and Y. Lakhnech, "Automated security proofs for almost-universal hash for MAC verification," in *ESORICS*, vol. 8134, 2013, pp. 291–308.

[43] A. J. Malozemoff, J. Katz, and M. D. Green, "Automated analysis and synthesis of block-cipher modes of operation," *IACR Cryptol. ePrint Arch.*, p. 774, 2014. [Online]. Available: http://eprint.iacr.org/2014/774

[44] V. T. Hoang, J. Katz, and A. J. Malozemoff, "Automated analysis and synthesis of authenticated encryption schemes," *IACR Cryptol. ePrint Arch.*, p. 624, 2015. [Online]. Available: http://eprint.iacr.org/2015/624

[45] K. Sieber, *The foundations of program verification*, 2013.

[46] A. Kfoury, "Hoare logic and variations: Probabilistic, relational, probabilistic+ relational," 2018. [Online]. Available: https://www.cs.bu.edu/faculty/kfoury/UNI-Teaching/CS512/AK_Documents/Hoare_Logic/main-post.pdf

[47] G. Barthe, B. Grégoire, Y. Lakhnech, and S. Zanella Béguelin, "Beyond provable security verifiable ind-cca security of oaep," in *Cryptographers' Track at the RSA Conference*, 2011, pp. 180–196.

[48] S. Zanella-Béguelin, G. Barthe, B. Grégoire, and F. Olmedo, "Formally certifying the security of digital signature schemes," in *S&P*, 2009, pp. 237–250.

[49] G. Barthe, D. Hedin, S. Z. Béguelin, B. Grégoire, and S. Heraud, "A machine-checked formalization of sigma-protocols," in *CSF*, 2010, pp. 246–260.

[50] G. Barthe, B. Grégoire, S. Heraud, and S. Z. Béguelin, "Computer-aided security proofs for the working cryptographer," in *Annual Cryptology Conference*, 2011, pp. 71–90.

[51] D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry, "Random oracles in a quantum world," in *ASIACRYPT*, 2011, pp. 41–69.

[52] G. Bana and H. Comon-Lundh, "A Computationally Complete Symbolic Attacker for Equivalence Properties," in *CCS*, 2014, pp. 609–620.

[53] D. Baelde, S. Delaune, C. Jacomme, A. Koutsos, and S. Moreau, "An Interactive Prover for Protocol Verification in the Computational Model," in *S&P*, 2021, pp. 537–554.

[54] F. Song, "A note on quantum security for post-quantum cryptography," in *Post-Quantum Cryptography*, 2014, pp. 246–265.

[55] G. Barthe, X. Fan, J. Gancher, B. Grégoire, C. Jacomme, and E. Shi, "Symbolic proofs for lattice-based cryptography," in *CCS*, 2018, p. 538–555.

[56] G. Barthe, M. Daubignard, B. Kapron, and Y. Lakhnech, "Computational indistinguishability logic," in *CCS*, 2010, p. 375–386.