Probabilistic Model–Checking of Quantum Protocols (Extended Abstract)

Simon Gay [1]

Department of Computing Science
University of Glasgow

Rajagopal Nagarajan [2,4] Nikolaos Papanikolaou [3,4]

Department of Computer Science
University of Warwick

1 Introduction

In the 1980s it was first realized that quantum–mechanical phenomena can be exploited directly for the manipulation, storage and transmission of information. The discovery of quantum algorithms for prime factorization [18] and unstructured search [7], which outperformed the best classical algorithms for these tasks significantly, opened up new vistas for computer science and gave an initial thrust to the emerging field of quantum computation. To implement a quantum algorithm, however, a large scale quantum computer is necessary and such a device has yet to be built. Research in *quantum information,* on the other hand, has shown that quantum effects can be harnessed to provide efficient and highly secure communication channels, which can be built using current technology. Entangled quantum states, superpositions and quantum measurement are among the characteristics of the subatomic world which nature puts at our disposal; these and related phenomena enable the development of novel techniques for computation and communication with no rival in classical computing and communication theory.

The focus in this paper is on communication protocols involving the use of such phenomena. Quantum protocols have particularly important applications in cryptography. Several quantum protocols have been proposed for cryptographic tasks such as oblivious transfer, bit commitment and key distribution [8,13]. The BB84 protocol for quantum key distribution [2,14], which allows two users to establish a common secret key using a single quantum channel, has been shown to be unconditionally secure against all attacks [11]. Other quantum protocols include procedures for

[1] Email: `simon@dcs.gla.ac.uk`

[2] Email: `biju@dcs.warwick.ac.uk`

[3] Email: `nikos@dcs.warwick.ac.uk`

[4] R. Nagarajan and N. Papanikolaou are supported by EPSRC grant GR/S34090 and the EU Sixth Framework Programme (Project SecoQC: *Development of a Global Network for Secure Communication based on Quantum Cryptography*).

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: `www.elsevier.nl/locate/entcs`

superdense coding [4], teleportation [3] and quantum error correction [19]. We assume that the reader is familiar with the basic concepts of quantum computing, as presented in [8,13].

We argue that detailed, automated analyses of protocols such as these facilitate our understanding of complex quantum behaviour and enable us to construct valuable proofs of correctness. Such analyses are especially important to manufacturers of commercial devices based on such protocols; the actual security of commercial quantum cryptographic systems, for example, is worth an in–depth investigation. Communication protocols have always been under scrutiny by computer scientists, who have developed numerous techniques for analysing and testing them, including process algebras, formal specification languages and automated verification tools. Automated verification techniques, such as *model-checking* and *theorem proving,* are frequently targeted at protocols and have been used to detect faults and subtle bugs. For instance, the FDR model-checker allowed Gavin Lowe to uncover a flaw in the Needham–Schroeder security protocol [17]. Although current model-checkers were developed primarily for the analysis of classical systems, we have found ways of using them to model quantum behaviour. To account for the probabilism inherent in quantum systems, we have chosen to use a *probabilistic* model–checker, in particular, the PRISM tool developed at the University of Birmingham [16].

PRISM is an acronym for *probabilistic symbolic model checker,* and is designed for modelling and validating systems which exhibit probabilistic behaviour. A tool such as PRISM computes the probability with which a system model σ satisfies a temporal formula Φ, i.e. the value of $\mathrm{P}_{\sigma,\Phi} = \mathrm{Pr}\{\sigma \models \Phi\}$ for given σ and Φ. The models catered for by PRISM may incorporate specific probabilities for various behaviors and so may the formulas used for verification. The application of probabilistic model–checking to quantum systems is entirely appropriate, since quantum phenomena are inherently described by random processes; to reason about such phenomena one must account for this.

PRISM uses a built–in specification language based on Alur and Henzinger's REACTIVE MODULES formalism (see [9,16] for details). Internally, a PRISM model is represented by a *probabilistic transition system.* The probabilistic temporal logic PCTL [5] is used as the principal means for defining properties of systems modelled in PRISM.

2 Fundamental Techniques

In order to use a classical probabilistic model–checker to verify quantum protocols, we need to model the quantum states that arise in a given protocol, and the effect of specific quantum operations on these states. PRISM itself only allows positive integer and boolean variables to be used in models. So how can we model the states of quantum systems, and the quantum operations arising in protocols, using only classical data types and arithmetic?

Single qubits can be in a superposition of two states, while classical variables can only take on a single value in any given state. The coefficients of these states can be any two complex numbers whose moduli squared sum to unity, and there is an uncountable infinity of these; of course, PRISM can only work with a finite state space. Furthermore, quantum systems consisting of many qubits can be in entangled states, which, unlike classical systems, cannot be decomposed into products of individual states. What is needed, therefore, is a means of representing quantum states fully and consistently, in a form that PRISM can handle.

Of all the possible quantum states of an n–qubit system, we identify the finite set of states which arise by applying the operations CNot, Hadamard (H), and $\sigma_0, \sigma_1, \sigma_2, \sigma_3$ to input states. We confine our analyses to protocols that involve *only* this restricted set of operations. At present, determining which states belong to this set is done manually, but we are considering ways of automating this.

A protocol such as superdense coding, which we will discuss in Section 3.1, can be expressed as a step-by-step interaction with a *two–qubit system.* In order to model the states of 2– and 3–qubit systems, the quantum operators and the measurements which arise in this and related protocols such as teleportation, we have developed a code generation tool called PRISMGEN. This tool generates a PRISM code fragment, or *module,* in which each quantum state is represented by a unique positive integer. Every quantum operator used in a particular protocol is coded as a set of deterministic transitions from one quantum state to another. PRISMGEN calculates these transitions by multiplying the unitary matrix, which corresponds to a particular operator, with each quantum state vector of interest. A measurement is modelled by a set of probabilistic transitions, leading to the various possible outcomes with equal probability. For simplicity, we have only considered states whose measurement outcomes are all equiprobable, although PRISM does allow us to model the more general case.

From the overall state space for a two–qubit system, a certain subset is closed under the CNot, Hadamard and Pauli operations. This subset consists of 4 states corresponding to the four basis vectors, 12 states which are sums of two basis vectors, and 8 states which are sums of all four basis vectors.

Proposition 2.1 *The above set of 24 states is closed under the* CNot*, Hadamard and Pauli operations.*

A proof of Proposition 2.1 is given in the full paper.

Our PRISMGEN tool enumerates these states and calculates the transitions corresponding to the various operations. The resulting PRISM module can be included as part of any model which involves measurements and the application of these operations to a system of two qubits. The situation with a system of three qubits is similar. We have developed a 3–qubit version of PRISMGEN, which gives us the ability to model protocols such as those for quantum teleportation and quantum error correction.

3 Illustrative Examples

We have been able to model a certain number of quantum protocols using the aforementioned techniques. These include: (1) superdense coding, which is a procedure for encoding pairs of classical bits into single qubits; (2) quantum teleportation, which allows the transmission of a quantum state without the use of an intervening quantum channel; and (3) quantum error correction, namely the qubit flip code, which corrects a single bit flip error during transmission of quantum bits. The source files for the models in this section are available online from http://go.warwick.ac.uk/nikos/research/.

In this extended abstract, only our analysis of superdense coding is presented.

3.1 Superdense Coding

The simplest quantum protocol which we will use to illustrate our techniques is the superdense coding scheme [4]. This scheme makes it possible to encode a pair of classical bits on a single qubit. With superdense coding, a quantum channel with a capacity of a single qubit is all that is necessary to transmit twice as many bits as a serial classical channel. Superdense coding is essentially a computation on a two–qubit system; therefore, the PRISM model of this protocol uses the 2–qubit version of PRISMGEN. We begin with a description of the protocol, and proceed to show how it is modelled and verified with PRISM.

The setting for superdense coding involves two parties, conventionally named Alice and Bob, who are linked by a quantum channel and share a pair of entangled qubits. The objective is for Alice to communicate the binary number xy — henceforth termed the *message* and denoted by (x, y), with $x, y \in \{0, 1\}$ — by transmitting a single qubit to Bob. The superdense protocol takes advantage of the correlations between qubits P_1 and P_2, which are in an entangled quantum state. Alice essentially influences this state in such a way that Bob's measurement outcome matches the message of her choice. The superdense coding protocol is as follows.

(i) Two qubits, P_1 and P_2, are placed in an entangled state using the Hadamard and CNot operations. Alice is given P_1, and Bob is given P_2.

(ii) Alice selects a message, (x, y), and applies the ith Pauli operator, σ_i, to P_1, where $i = y + x(2 + (-1)^y)$. She transmits this particle to Bob.

(iii) Bob applies the CNot gate from P_1 to P_2, and then he applies the Hadamard gate to the former.

(iv) Bob measures the two particles, thus obtaining a pair of classical bits, (x', y'). If no disturbance has occurred, this pair of bits will match the original message, i.e. $(x', y') = (x, y)$.

The model of superdense coding consists of four PRISM modules. Of these four, one module is generated automatically by PRISMGEN and describes the possible states of the two qubits. There is a module specifying Alice's actions, and similarly one for Bob's. Before we examine the workings of this model in detail, consider the following observations, which highlight the capabilities of PRISM. In the PRISM model, Alice's first action is to select one of the four possible messages (represented by the integers 0, 1, 2, 3); each message has an equal probability, $\frac{1}{4}$, of being chosen. This is an assumption we made when constructing this model, but it is possible to specify different respective probabilities for the four choices. Another point worth noting is that, depending on which message is chosen, the protocol proceeds in one of four distinct ways; PRISM actually considers *all* these possibilities when testing the validity of a property. This is precisely why we advocate the use of model-checking for these analyses, as opposed to simulation of quantum protocols, proposed elsewhere; simulators only treat one of several possible executions at a time.

PRISM interprets the superdense coding model as a probabilistic transition system, which can be depicted as a graph. The nodes in the graph correspond to the internal state numbers which PRISM assigns to each step in the protocol. Each internal state number corresponds to a tuple with the states of all variables in a particular model. An illustration of this graph and the details of the internal state numbers will be included in the full paper.

The quantum state of the two-qubit system is represented by the variable `state` in the PRISM model. When Bob has finished his measurement, and the dense coding protocol terminates, one of 4 final states is reached (each representing a distinct possibility in the computation). The property required for verification must be expressed in terms of the final state. When the dense coding protocol terminates, Bob's measurement result, i.e. the pair of classical bits (x', y'), must match Alice's original choice (x, y). This requirement is expressed using PCTL, as follows:

$$\mathrm{P} \geqslant 1 \; [\; \mathbf{true} \, \mathcal{U} \; ((\texttt{protocol\_finished}) \;\wedge\; (\texttt{result} = \texttt{msg})) \;] \tag{1}$$

The PCTL formula in (1) stipulates that the probability of Bob's result matching Alice's choice is 1. Model–checking with PRISM confirms that this property holds (i.e. this property is **true** for all executions of the model). We have thus proven, using the PRISM model–checker, that the dense coding protocol always succeeds in transmitting two classical bits using a single qubit. Clearly, this is not difficult to prove by hand; however, we have used dense coding as a simple demonstration of our approach.

4 Challenges and Future Prospects

We have demonstrated our approach to the analysis of quantum communication protocols using a simple examples. There is significant scope for future work, ranging from improvements to our current code–generation techniques, to the automated verification of larger systems, such as quantum cryptographic devices.

At present we explicitly construct state spaces and transition tables for systems involving up to three qubits and the H, CNot and σ_i operators. We have informally reached the conclusion that, for any number of qubits, there is a finite set of states which is closed under these operators. It is not directly obvious how many states these are, but this could be established computationally. There is a mathematical framework called the *stabilizer formalism,* which could be used to calculate these states. Investigating this formalism and its implications could lead to a more efficient model checking for protocols; it is already known that stabilizer circuits can be efficiently simulated by a classical computer [1].

The guarded transitions of PRISM's modelling language make it awkward to express some basic control structures such as sequencing. Each PRISM module typically requires a variable which acts as a program counter and must be explicitly incremented in each transition. We intend to develop automatic translations from the high–level process calculus CQP [6] into PRISM's native language. Combining such a specification formalism for protocol models with a logic for defining properties will allow us to verify quantum protocols at a higher level.

Our ultimate aim is to construct models of larger systems which combine quantum and classical components, or which combine more than one quantum protocol. For example, we are working on augmenting an existing model [15] of the BB84 key–distribution protocol with descriptions of authentication, secret–key reconciliation, and privacy amplification protocols [8]. As PRISM allows probabilities of particular events to be calculated directly, we can obtain numerical values of probability, such as those that arise in mathematical analyses of security; we have taken advantage of this capability in our existing model of BB84. More complex protocols generally involve larger numbers of qubits, leading to ever greater state spaces for verification.

5 Conclusions

We have established, for the first time, techniques for analyzing and verifying quantum communication systems. Our key contributions are the development of a general approach to modelling the state space of systems of several qubits, and the introduction of techniques for defining properties of quantum protocols in the logic PCTL. We have illustrated our approach by modelling and verifying three example protocols (focusing on superdense coding only here) using PRISM. Although these examples are simple, they are important building blocks of the theory of quantum communication. Having established fundamental and general techniques for formal verification of quantum protocols, we are in a strong position to carry out end–to–end verifications of larger systems, such as those used for quantum cryptography.

References

[1] Aaronson, S. and D. Gottesman, *Improved simulation of stabilizer circuits* (2003), available at arXiv.org. Record: quant-ph/0406196.

[2] Bennett, C. H. and G. Brassard, *Quantum cryptography: Public key distribution and coin tossing*, in: *Proceedings of International Conference on Computers, Systems and Signal Processing*, 1984.

[3] Bennett, C. H., G. Brassard, C. Crépeau, R. Jozsa, A. Peres and W. K. Wootters, *Teleporting an unknown quantum state via dual classical and Einstein–Podolsky–Rosen channels*, Physical Review Letters **70** (1993), pp. 1895–1899.

[4] Bennett, C. H. and S. J. Wiesner, *Communication via one- and two-particle operators on Einstein–Podolsky–Rosen states*, Physical Review Letters **69** (1992), pp. 2881—2884.

[5] Ciesinski, F. and M. Größer, *On probabilistic computation tree logic.*, in: *Validation of Stochastic Systems*, 2004, pp. 147–188.

[6] Gay, S. and R. Nagarajan, *Communicating quantum processes*, in: *POPL '05: Proceedings of the 32nd ACM Symposium on Principles of Programming Languages, Long Beach, California*, 2005.

[7] Grover, L. K., *A fast quantum mechanical algorithm for database search*, in: *Proc. 28th Annual ACM Symposium on the Theory of Computing (STOC)*, 1996, pp. 212–219.

[8] Gruska, J., "Quantum Computing," McGraw–Hill International, 1999.

[9] Kwiatkowska, M., G. Norman and D. Parker, *Modelling and verification of probabilistic systems*, in: P. Panangaden and F. V. Breugel, editors, *Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems*, American Mathematical Society, 2004 Volume 23 of CRM Monograph Series.

[10] Mateus, P. and A. Sernadas, *Reasoning about quantum systems*, in: *Proceedings of the Ninth European Conference on Logics in Artificial Intelligence (JELIA'04)*, number 3229 in Lecture Notes in Artificial Intelligence (2004).

[11] Mayers, D., *Unconditional security in quantum cryptography*, Journal of the ACM **48** (2001), pp. 351—406.

[12] Nagarajan, R. and S. Gay, *Formal verification of quantum protocols* (2002), available at arXiv.org. Record: quant-ph/0203086.

[13] Nielsen, M. A. and I. L. Chuang, "Quantum Computation and Quantum Information," Cambridge University Press, 2000.

[14] Papanikolaou, N., *Introduction to quantum cryptography*, ACM Crossroads Magazine **11.3** (2005), pp. 10—16.

[15] Papanikolaou, N., "Techniques for Design and Validation of Quantum Protocols," Master's thesis, Department of Computer Science, University of Warwick (2005).

[16] Parker, D., G. Norman and M. Kwiatkowska, PRISM *2.0 users' guide* (2004).

[17] Ryan, P., S. Schneider, M. Goldsmith, G. Lowe and B. Roscoe, "Modelling and Analysis of Security Protocols," Pearson Education, 2001.

[18] Shor, P., *Algorithms for quantum computation: discrete logarithms and factoring*, in: *Proceedings of 35th Annual Symposium on Foundations of Computer Science* (1994).

[19] Steane, A. M., *Quantum computing and error correction*, in: A. Gonis and P. Turchi, editors, *Proceedings of the NATO Advanced Research Workshop* (2000), pp. 284—298.