Model–Checking Quantum Key Distribution: Techniques and Results

Nick Papanikolaou[1]

Joint work with Simon Gay[2] and Rajagopal Nagarajan[1]

[1]Department of Computer Science
University of Warwick

[2]Department of Computing Science
University of Glasgow

FOCS / Q-Day II, 8/12/05

Outline

Motivation

- Practical systems for QKD are already available commercially (viz. `www.magiqtech.com`, `www.idquantique.com`).
- The unconditional security proof of QKD does not take into account implementation–level details; it relies only on information–theoretic arguments.
- We are in favour of a more practical approach, which is at a closer level to implementation: **probabilistic model–checking**.
- We will demonstrate this approach with an elementary analysis of the BB84 protocol for QKD.
- We have already extended the approach to other protocols.

Quantum Key Distribution

- **Key distribution** is the process of establishing a common secret $k \in \{0,1\}^N$ known as the **key**, between two users (Alice and Bob).
- Classical key distribution is, at best, **computationally secure**.
- QKD is **unconditionally secure** against all attacks permitted by quantum mechanics (Mayers, 1996).
- Several protocols have been proposed for QKD:
  - **BB84 (Bennett and Brassard, 1984)**
  - B92 (Bennett, 1992)
  - E91 (Ekert, 1991)

The BB84 Protocol

1. Alice generates a random stream of qubits in the basis states of either the **standard basis** or the **Hadamard basis**. She sends all the qubits to Bob.
2. Bob chooses one of two observables M_s, M_h and measures each qubit received. He stores the outcomes.
3. Alice and Bob compare their choices of bases and observables. All mismatches are discarded.

- To model this protocol, we store only 1 qubit at a time and repeat the process.
- The state space for this protocol is the set

$$S = \left\{ |0\rangle, |1\rangle, \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right), \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) \right\}$$

where S is closed under the H unary operator and the two measurement observables M_s and M_h.

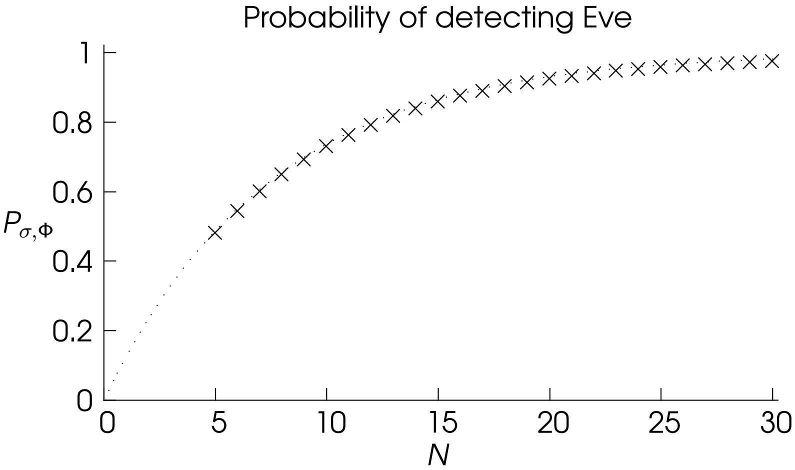
Probabilistic Model Checking

- A **probabilistic model checker** is designed to allow the verification of concurrent systems with probabilistic behaviour.
  - **PRISM (Kwiatkowska et al., 2001)**
  - ProbVerus (Clarke et al., 1999)
  - ProbUSM (Baier et al., 2005)
- A PRISM model consists of agents performing named actions with specified probabilities.
- A PRISM property is an expression in Probabilistic Computation Tree Logic (PCTL).
- For a given model σ and temporal formula ϕ, PRISM computes $\Pr(\sigma \models \phi)$.

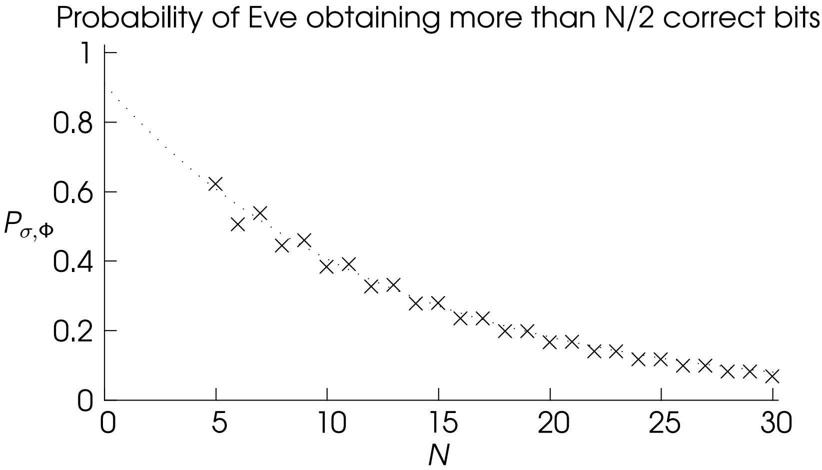
PRISM Models of BB84

- We have used PRISM to create a model of the basic BB84 protocol. With PRISM we have computed:
  - the probability P_{det} of detecting an eavesdropper when N qubits are transmitted; and
  - the probability $P_{>1/2}$ that the eavesdropper obtains more than half the originally transmitted bit values by measurement.
- The model has a single parameter, the number N of qubits transmitted by Alice to Bob over the quantum channel.
- We have computed the probabilities P_{det} and $P_{>1/2}$ for N ranging from 5 to 30.

Intercept–Resend Eavesdropping: P_{det}



Intercept–Resend Eavesdropping: $P_{>1/2}$



Developing a General Framework

- Our programme is **to develop a general, high–level framework** for modelling and analysing quantum protocols using model checking.
- We are developing a **code generation tool**, PRISMGEN, which generates finite models for this purpose.
- We aim to combine our formal verification framework with a high–level specification language, in particular **CQP** (Gay and Nagarajan, 2005).
    - Problem is to build models for M–qubit systems, whose state spaces grow exponentially with M.
    - By using code generation, we can abstract away from PRISM's low–level language and provide high–level protocol primitives.

Generating Models of State Spaces for Protocols

- The BB84 model only stores 1 qubit at a time.
- **General technique:** to identify the finite set/group of quantum states which are closed under the specific set of operations used in a quantum protocol.
- In **quant-ph/0504007** we show how this idea is applied to simple examples: superdense coding, quantum teleportation, and a simple quantum error correction circuit.
- **PRISMGEN:** tool for generating a PRISM agent ("module") representing an M–qubit system and the effect of basic operations $H, CNot, \sigma_x, \sigma_y, \sigma_z$.
- We have had success to date for $M = 2$ and $M = 3$ qubits - adequate for simple examples.

Review

- We have presented a basic analysis of the **BB84** protocol.
- We have discussed the use of the **PRISM** in this context.
- We have considered the problem of **generating state spaces** for quantum protocols.

- We have **not** presented the precise nature of the models here.
- We have **not** discussed the algorithm for generating a unique state space.
- We have **not** considered the inherent limitations of the approach.

For Further Reading

- NAGARAJAN, PAPANIKOLAOU, BOWEN, AND GAY
  An Automated Analysis of the Security of Quantum Key Distribution.
  In *Proceedings of SECCO'05, San Francisco*, August 2005.
- GAY, S., NAGARAJAN, R., AND PAPANIKOLAOU, N.
  Probabilistic Model–Checking of Quantum Protocols.
  Preprint quant-ph/0504007, available at `www.arxiv.org`.
- GAY, S. AND NAGARAJAN, R.
  Communicating Quantum Processes.
  In *POPL '05: Proceedings of the 32nd ACM Symposium on Principles of Programming Languages, Long Beach, California*, January 2005.