A Genetic Algorithm to Analyze the Security of Quantum Cryptographic Protocols

Walter O. Krawec

walter.krawec@gmail.com

Iona College
Computer Science Department
New Rochelle, NY USA

IEEE WCCI July, 2016

Quantum Key Distribution (QKD)

1. Allows two users - Alice (A) and Bob (B) - to establish a shared secret key
2. Secure against an all powerful adversary
   - Does not require any computational assumptions
   - Attacker bounded only by the laws of physics
   - Something that is not possible using classical means only
3. Accomplished using a *quantum communication channel*

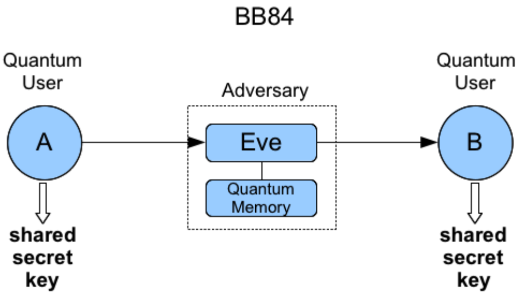


Figure: Typical QKD Setup

QKD in Practice

1. Quantum Key Distribution is here already
2. Several companies produce commercial QKD equipment
    1. MagiQ Technologies in NY
    2. id Quantique in Geneva
    3. SeQureNet in Paris
    4. Quintessence Labs in Australia
3. Have also been used in various applications:
    1. In 2007, QKD was used to transmit ballot results for national elections in Switzerland
    2. Has also been used to carry out bank transactions

Quantum Key Distribution

1. QKD Protocols typically operate by first having A and B communicate using *qubits*.
2. Several iterations of this pass defining the *quantum communication stage*
3. Results in A and B each holding a *raw key*, a string of classical bits that is:
    - Partially correlated
    - Partially secret

Quantum Key Distribution (continued)

1. Due to certain properties of quantum communication, E's attack introduces noise into the quantum channel
2. The amount of noise correlates directly with the maximal information E holds on the raw key
3. If the noise level is "too high" then A and B must abort
4. Otherwise, if it is lower than some security threshold τ_Q, they may distill a secure secret key (using *error correction* and *privacy amplification*)
5. Question: **What is τ_Q?**

Noise Threshold

1. While τ_Q is known for many protocols (e.g., for BB84 it is 11% [1]), many newer protocols have no such bound or only lower-bounds.
2. Especially problematic are two-way protocols which hold numerous practical advantages (important, since QKD protocols are available with current-day technology!)

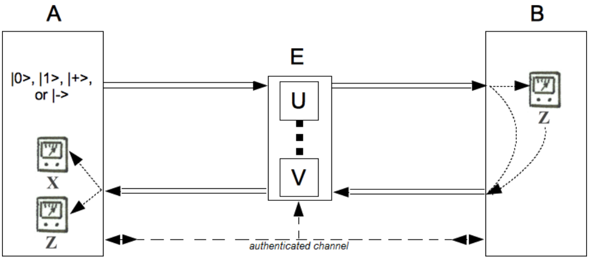


Figure: A Two-Way QKD Protocol

Our Goal

1. We propose a real-coded GA which searches over the space of E's attack operators to find an upper-bound on τ_Q for general QKD protocols both one-way and two-way (and n-way)
2. Useful for protocols where no rigorous proof of security exists
3. Lower-bounds are often easier to prove mathematically; this tool gives researchers an upper-bound
4. Also: useful tool for researchers to test a new protocol before going into mathematical details (e.g., to see if it is secure)
5. Can be used to quickly test (and discover) new conjectures in quantum cryptography

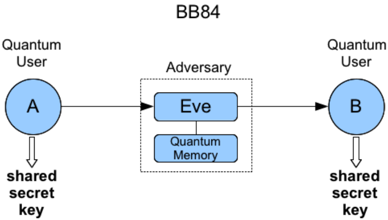
Related Work

1. Numerous authors have applied evolutionary techniques to problems in quantum computation [2, 3, 4, 5]
2. Most deal with finding operators (algorithms) to solve certain computational problems
3. In an extended abstract [6] we first proposed the idea of using a GA to analyze QKD protocols
4. In this paper, we extend this technique to work with more general QKD protocols and perform a more thorough analysis; we also add new abilities to the algorithm.
5. To our knowledge, we are the first to apply evolutionary techniques successfully to analyze the security of QKD protocols according to state-of-the-art definitions of QKD security

Background

Quantum Key Distribution

1. A QKD protocol first performs the *quantum communication (QC) stage*
2. A and B communicate by passing *qudits* to one-another over several iterations
3. E captures and "probes" each passing qudit (no-cloning!)
4. Two events: A and B use an iteration for *raw key distillation* or *parameter estimation* (how noisy is the channel?)
   - Announced publicly after the fact...



Quantum Key Distribution

1. After the QC stage, A and B have a classical *raw key*...
2. ... and E has a large quantum system in her perfect quantum memory.
3. If the noise is "small enough" the users run Error Correction and Privacy Amplification (using a public authenticated classical channel)
4. Takes a raw key of N-bits and outputs a secret key of size:

   $$\ell(N) \leq N$$

   (possibly $\ell(N) = 0$ if E has too much information).
5. **Question**: Given a noise rate of Q, what is $\ell(N)$ and when is it zero?

Modeling E's attack

1. We consider *collective attacks* (usually good enough!)
2. Let K be the number of times a qudit passes through E in a single iteration (usually $K = 1$ or 2).
3. Then, E's attack is a collection of K *unitary operators* (without loss of generality, finite dimension) $\{U_1, \cdots, U_K\}$
   - U_i is unitary if $U_i \cdot U_i^* = I$.
4. These operators act on the traveling qudit and also E's private quantum memory

QKD Key Rate

1. Ultimately, a QKD protocol may be modeled mathematically as a (possibly large) matrix with complex entries (a *Density Operator*) "ρ"
2. It was shown in [1] that:

$$\#\text{ secret bits} = \ell(N) \approx N \cdot r,$$

where:

$$r = \inf_U r(U) = \inf(\overbrace{S(AE) - S(E) - H(A|B)}^{r(U)}) \leq 1,$$

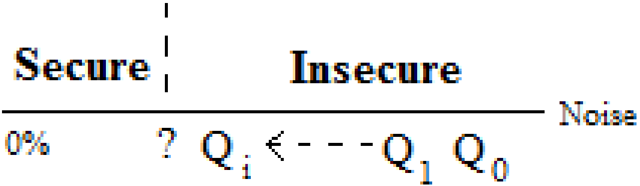
and $S(AE)$ (resp. $S(E)$) is the von Neumann entropy of the Density Operator modeling A and E's (resp. E's) system.
3. To compute $\ell(N)$ need the von Neumann entropy of ρ which means finding the eigenvalues of ρ

Algorithm Idea

1. We will search over the space of all attack operators $U = \{U_i\}$
2. Try to find U that induces a minimal amount of noise (i.e., it is not very invasive as far as A and B are concerned), yet this same U should cause $R(U) = 0$.
3. Once such an operator is found, it may be concluded that the protocol in question cannot possibly withstand noise levels higher than that induced by U (the infimum will be even smaller).

Algorithm Idea

Our algorithm, therefore, finds upper-bounds on the maximally tolerated noise threshold of a given QKD protocol



Solution Representation

Some Basic Quantum Terminology

1. A quantum system is modeled as a vector $|\psi\rangle$ in a complex vector space
2. Example:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$|+\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \qquad |e_2\rangle = \begin{pmatrix} .2 \\ .01 \\ -.07 \\ 0 \end{pmatrix}$$

3. If $|\psi\rangle$ and $|\phi\rangle$ are two vectors, we write:

$$\langle\psi|\phi\rangle$$

to be their inner-product.

Some Basic Quantum Terminology

1. If $|\psi\rangle$ is an *n*-dimensional vector representing a quantum system and $|\phi\rangle$ is an *m*-dimensional vector representing a different system...
2. ... then we model the joint state as:

$$|\psi, \phi\rangle = |\psi\rangle \otimes |\phi\rangle$$

which is an $n \cdot n$ dimensional vector.

Solution Representation

1. E's attack is a collection of unitary operators $\{U_1, \cdots, U_K\}$ such that $U_i \cdot U_i^* = I$.
2. Requires $O(n^2)$ variables to describe
3. But: we don't need the entire operator, we only need to know its action on certain "basis states"

Solution Representation

1. Example Round 1, $T = 2$: We only need U_1's action on **basis** states: $|0,0\rangle$, $|1,0\rangle$:

$$U_1 |0,0\rangle = |0, e_0^1\rangle + |1, e_1^1\rangle$$
$$U_1 |1,0\rangle = |0, e_2^1\rangle + |1, e_3^1\rangle$$

   (Can be generalized to $T > 2$; i.e., $|i, 0\rangle$)

2. Each $|e_i^1\rangle$ is a complex vector (dimension specified by user)
3. Unitarity of U_1 forces the condition:

$$\langle e_0^1|e_0^1\rangle + \langle e_1^1|e_1^1\rangle = 1$$
$$\langle e_2^1|e_2^1\rangle + \langle e_3^1|e_3^1\rangle = 1$$
$$\langle e_0^1|e_2^1\rangle + \langle e_1^1|e_3^1\rangle = 0$$

Solution Representation: Round 1

$$U_1 |0, 0\rangle = |0, e_0^1\rangle + |1, e_1^1\rangle \qquad U_1 |1, 0\rangle = |0, e_2^1\rangle + |1, e_3^1\rangle$$

1. Let T be dimension of *Transit Space* (e.g., $T = 2$) and d_1 the dimension of E's round 1 quantum memory (upper-bounded by T^2)
2. A candidate solution for round 1 is a collection of T vectors:

$$\begin{aligned} \mathcal{G}_0^1 &= (g_0^1, g_1^1, \cdots, g_{T-1}^1) \\ \mathcal{G}_1^1 &= (g_T^1, g_{T+1}^1, \cdots, g_{2T-1}^1) \\ &\vdots \end{aligned}$$

   with each g_i^1 consisting of d_1 random complex numbers
3. Clearly, this does not satisfy the required unitary conditions...

Solution Representation: Round 1

1. Next, run the Gram-Schmidt process to orthogonalize the vectors:

$$
\begin{aligned}
\mathcal{G}_0^1 &\rightsquigarrow \mathcal{F}_0^1 = (f_0^1, f_1^1, \cdots, f_{T-1}^1) \\
\mathcal{G}_1^1 &\rightsquigarrow \mathcal{F}_1^1 = (f_T^1, f_{T+1}^1, \cdots, f_{2T-1}^1) \\
&\vdots
\end{aligned}
$$

Solution Representation: Example

- Let $T = 2$, then we need states (vectors) $|e_i^0\rangle$:

$$U_1 |0,0\rangle = |0, e_0^1\rangle + |1, e_1^1\rangle \qquad U_1 |1,0\rangle = |0, e_2^1\rangle + |1, e_3^1\rangle$$

  such that:

$$\begin{aligned} \langle e_0^1 | e_0^1 \rangle + \langle e_1^1 | e_1^1 \rangle &= 1 \\ \langle e_2^1 | e_2^1 \rangle + \langle e_3^1 | e_3^1 \rangle &= 1 \\ \langle e_0^1 | e_2^1 \rangle + \langle e_1^1 | e_3^1 \rangle &= 0 \end{aligned}$$

- We have orthonormal vectors:

$$\begin{aligned} \mathcal{F}_0^1 &= (f_0^1, f_1^1) \\ \mathcal{F}_1^1 &= (f_2^1, f_3^1) \end{aligned}$$

Solution Representation

- To evolve the entire unitary operator U_1 would require $(T \cdot d_1)^2$ variables
- Instead, we require $2d_1 \cdot T^2$
- If $T = 2$ and $d_1 = 4$ (common values), then we have 32 variables (as opposed to 64)

Solution Representation: Round 2

1. The second round attack (i.e., U_2) is a little more involved
2. It acts on the transit space, E's last memory "block" (dimension d_1) and a new memory block of dimension d_2.
3. We fix a basis for E's last-used memory ancilla, **based on U_1's action**, and write U_2's action on basis states of the form $|i, j, 0\rangle$ where $i = 0, 1, \cdots T - 1$, and $j = 0, 1, \cdots, d_1 - 1$.
4. We then follow the process described above

Solution Representation: Number of Variables

Evolve Entire Unitary Operators:

- Round 1: U_1 requires $T^2 \cdot d_1^2$ variables
- Round 2: U_2 requires $T^2 \cdot d_1^2 \cdot d_2^2$ variables
- Example: $T = 2, d_1 = 4, d_2 = 64$ (most powerful attack)
- Requires $64 + 262144 = \boxed{262,208}$ variables

Evolve Unitary Description (Our Method):

- Round 1: requires $2T^2 d_1$ variables
- Round 2: requires $2T^2 d_1^2 d_2$ variables
- Example: $T = 2, d_1 = 4, d_2 = 64$
- Requires $32 + 8192 = \boxed{8,224}$ variables

The Algorithm

Genetic Operators

1. A candidate solution is a collection of complex vectors $\mathcal{G}_i^j$
2. Initial population generated by choosing real and imaginary parts randomly in the interval $[-2, 2]$.
3. Crossover is simple one-point crossover (choosing a different crossover point for each vector $\mathcal{G}_i^j$
4. Mutation will alter 25% of all elements by adding a small $\epsilon \in [-1/10, 1/10]$ to real and imaginary parts
5. After any genetic operation, the vectors $\mathcal{F}$ are reconstructed using the G.S. process from which the $|e_i^j\rangle$ states are derived.

Algorithm: Input

1. The algorithm takes as input a description of the QKD protocol including:
   - How is a key-bit created? (1x)
   - How is the noise measured? (1x or more)
2. Description created in a custom-made language...

Algorithm: Example Input (BB84 [7])

```
create space (AKey:2, BKey:2, Basis:2, Transit:2, Eve1:4)

with prob .25 prepare (Basis=0, Transit=0, AKey=0)
elsewith prob .25 prepare (Basis=0, Transit=1, AKey=1)
elsewith prob .25 prepare (Basis=1, Transit=0, AKey=0)
elsewith prob .25 prepare (Basis=1, Transit=1, AKey=1)
endwith

apply conditional op H to Transit if (Basis=1)

attack (Transit)

apply conditional op H to Transit if (Basis=1)

measure Transit save in BKey

trace out Transit

save as primary
```

Fitness

1. From the above descriptions and a candidate attack operator, **density operators** are constructed
2. From this, we may compute the noise level Q and the key rate:

$$R(U) = S(AE) - S(E) - H(A|B)$$

3. Goal is to find U which minimizes the noise (less invasive) and minimizes the key-rate (more information to E).
4. We use the fitness function:

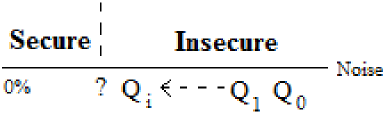
$$\text{fit}(U) = p_f(Q - \tau_Q)^2 + (1 - p_f)(R + .01)^2,$$

where τ_Q is a user-specified target noise rate (usually 0) and p_f is a weight (usually $1/2$).

The Algorithm

1. Create initial population
2. Take best-fit solution U; if $R(U) < 0$ then save noise level as $\hat{Q}$
3. Evolve next generation
4. Goto 2 until some stopping condition is met
5. Output $\hat{Q}$

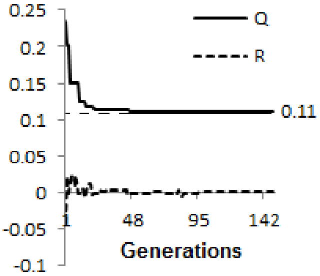
It is guaranteed that the given protocol cannot tolerate a noise level of $\hat{Q}$



Evaluation

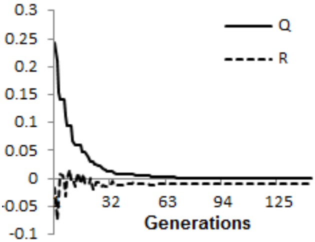
BB84

1. First test: BB84 [7]
2. Well known that the tolerated error rate is 11%
3. Algorithm Output (50 runs):
   - Best: 11.01%
   - Average: 11.07%
   - Standard deviation: 4.0×10^{-4}



BB84: Insecure Version

1. Tested an insecure version of BB84
2. Algorithm found a solution with little noise ($Q < 0.00087$) and a zero key-rate.
3. Thus, our algorithm can be used to quickly check if a protocol is secure.



B92

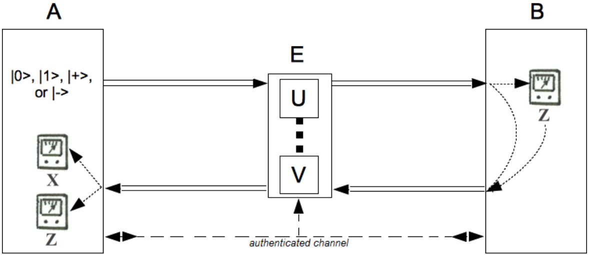
1. Next test: B92 [8] a minimal QKD protocol more sensitive to noise
2. Algorithm Output (50 runs):
   - Average: 7.73%
   - Standard deviation: 1.5×10^{-4}
3. Current best **lower-bound** is 6.5% [9]; actual tolerated threshold somewhere between these two results.
4. Often it is easier to prove rigorous lower-bounds; our analysis software provides upper-bounds

SARG04

1. SARG04 [10] an extended version of B92
2. Theoretical noise threshold: 9.68%
3. Algorithm Output (50 runs):
    - Average: 10.25%
    - Standard deviation: 3.5×10^{-4}

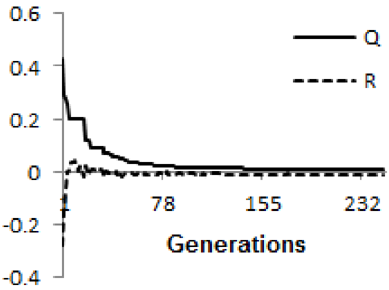
Two-Way: SQKD

1. We consider a new class of two-way QKD protocol: a semi-quantum protocol [11]
2. Theoretical lower-bound: 7.4%
3. Algorithm Output (50 runs):
    - Average: 8.7%
    - Standard deviation: 5.6×10^{-3}



Insecure SQKD

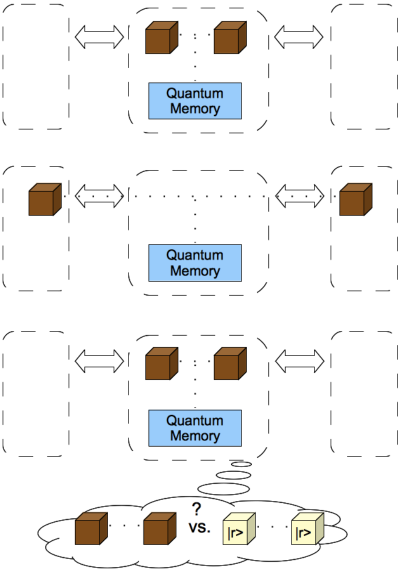
1. An SQKD protocol requires that the user B send a qubit in an exact state back to A under certain events
2. If we alter the protocol so that B sends a different state, the resulting protocol is insecure according to our algorithm
3. We verified this mathematically



Mediated QKD

1. Finally, we evaluated a mediated QKD protocol [12]
2. Requires the attacker to prepare qubits, send one to A another to B, measure the returning state, and send a classical message to the users
3. Thus, our algorithm must evolve a strategy that optimizes E's information, but also interacts with the two users meaningfully

Mediated QKD



Mediated QKD

1. Requires 7 different noise measurements
2. Algorithm successfully evolved an attack strategy which did not cause A and B to abort
3. Theoretical lower-bound: 10.8%
4. Algorithm Output (28 runs):
   - Average: 12.5%
   - Standard deviation: 2.59×10^{-2}

Future Work

Future Work

1. Different solution representation (e.g., gate-based)
2. Consider practical attacks
3. Different attack models (e.g., noisy quantum storage)
4. Also, multi-photon attacks and photon-losses

Thank you! Questions?

References I

Renato Renner, Nicolas Gisin, and Barbara Kraus.
Information-theoretic security proof for quantum-key-distribution protocols.
Phys. Rev. A, 72:012332, Jul 2005.

S. R. Hutsell and G. W. Greenwood.
Applying evolutionary techniques to quantum computing problems.
In *IEEE Congress on Evolutionary Computation (CEC 2007)*, pages 4081–4085, September 2007.

M. Lukac and M. Perkowski.
Evolving quantum circuits using genetic algorithm.
In *EH '02: Proceedings of the 2002 NASA/DoD Conference on Evolvable Hardware*, pages 177–185, 2002.

L. Spector.
Automatic Quantum Computer Programming: A Genetic Programming Approach.
Kluwer Academic Publishers, Boston, MA, 2004.

Walter O Krawec.
An algorithm for evolving multiple quantum operators for arbitrary quantum computational problems.
In *Proceedings of the 2014 conference companion on Genetic and evolutionary computation companion*, pages 59–60. ACM, 2014.

References II

Walter O Krawec.
Using evolutionary techniques to analyze the security of quantum key distribution protocols.
In *Proceedings of the 2014 conference companion on Genetic and evolutionary computation companion*, pages 171–172. ACM, 2014.

Charles H Bennett and Gilles Brassard.
Quantum cryptography: Public key distribution and coin tossing.
In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, volume 175. New York, 1984.

Charles H. Bennett.
Quantum cryptography using any two nonorthogonal states.
Phys. Rev. Lett., 68:3121–3124, May 1992.

Ryutaroh Matsumoto.
Improved asymptotic key rate of the b92 protocol.
In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 351–353. IEEE, 2013.

Antonio Acin, Nicolas Gisin, and Valerio Scarani.
Coherent-pulse implementations of quantum cryptography protocols resistant to photon-number-splitting attacks.
Phys. Rev. A, 69:012309, Jan 2004.

References III

Michel Boyer, D. Kenigsberg, and T. Mor.
Quantum key distribution with classical bob.
In *Quantum, Nano, and Micro Technologies, 2007. ICQNM '07. First International Conference on*, pages 10–10, 2007.

Walter O Krawec.
An improved asymptotic key rate bound for a mediated semi-quantum key distribution protocol.
Quantum Information and Computation, 16(9 & 10):0813–0834, 2016.