Evolutionary Algorithms for the Design of Quantum Protocols

Walter Krawec[1], Stjepan Picek[2], and Domagoj Jakobovic[3]

[1] Department of Computer Science and Engineering, University of Connecticut, Storrs CT, 06268, USA
[2] Cyber Security Research Group, Delft University of Technology, Mekelweg 2, Delft, The Netherlands
[3] Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia

Abstract. In this paper, we use evolutionary algorithm to evolve customized quantum key distribution (QKD) protocols designed to counter attacks against the system in order to optimize the speed of the secure communication. This is in contrast to most work in QKD protocols, where a fixed protocol is designed and then its security is analyzed to determine how strong an attack it can withstand. We show that our system is able to find protocols that can operate securely against attacks where ordinary QKD protocols would fail. Our algorithm evolves protocols as quantum circuits, thus making the end result potentially easier to implement in practice.

Keywords: Quantum Cryptography, Evolution Strategy, Quantum Simulator

1 Introduction

Quantum cryptography is a fascinating area of study allowing for the achievement of certain important communication tasks which ordinarily would be impossible through classical communication only. One prominent example of this is *quantum key distribution* (QKD) which permits the establishment of a secret key (a classical bit string which may be used for other cryptographic tasks such as message encryption) between two users A and B, which is secure against an all-powerful adversary (something impossible to achieve through classical communication alone). Beyond this theoretical advantage, *QKD protocols are currently a practical technology which has seen several real-world applications.* Indeed, numerous experimental groups have verified the correctness and applicability of QKD. Additionally, there are currently several companies producing commercial QKD equipment and new QKD networks being established worldwide. For a general survey of QKD, both the theory and practice, the reader is referred to [1].

One of the unique properties of quantum communication is that there is (assuming the protocol is correct) a direct correlation between the observed noise in a quantum channel and the maximal amount of information any adversary

could have gained on the information being sent. QKD protocols are able to operate successfully up until a certain noise threshold is reached (called a protocol's noise tolerance). Before this threshold is reached, a secure key can be distilled - however the process grows less efficient as the noise level increases.

Generally, QKD protocols are constructed and then analyzed mathematically to determine what channels they can operate over (e.g., what are their noise tolerance), and, furthermore, what their efficiency is for particular channels. As an example, the BB84 protocol [2] can work over a symmetric channel so long as the noise is less than 11% [3]. Other protocols exist each with their own noise tolerances (along with other advantages or disadvantages). However, these noise tolerance results generally only hold for symmetric channels - over asymmetric channels, this is not necessarily true! In fact, for certain channels (i.e., attacks against the protocol), none of the current existing protocols may provide optimal noise tolerances and communication efficiency. In this paper, we are interested in the problem of finding QKD protocols optimized to work over given quantum qubit channels so as to maximize the efficiency of the secret key distribution rate beyond what current state-of-the-art QKD protocols may be able to do over this same channel.

In particular, we envision a system whereby users of quantum communication technology may, after running standard quantum tomographic protocols to measure the noise in the quantum channel, insert these measurement results into our algorithm which will then construct a tailor-made QKD protocol specifically designed to counteract the noise in the quantum channel. This quantum tomographic protocol involves users simply sending and receiving quantum bits prepared in a variety of manners so as to produce a *noise signature* of the channel - i.e., a list of several important channel noise statistics which can be used to characterize, at least partially, the adversary's attack. Since adversarial attacks are, in the worst-case, the cause of the noise in the channel, our system is constructing protocols that counteract an all-powerful, quantum capable, adversary. Other applications of our approach may be to counter changes in operating conditions (e.g., changes in environmental conditions which alter the noise in, say, a free-space channel). Such a system may be eventually used to create a more efficient quantum secure communication network. Furthermore, our system will, in fact, give explicit instructions on how to operate quantum devices by providing to users quantum protocols as basic *quantum circuits*. Furthermore, the circuits produced will consist of gates from a user-specified gate set, thus our system can take into account the capabilities, and limitations, of user hardware.

Our approach will utilize evolutionary algorithms to discover optimized QKD protocols. Evolutionary algorithms have been used for some time with success to evolve quantum algorithms [4–6], usually being used to find quantum circuits that are more efficient than human-constructed versions. Some work using (simulated) quantum computers to run classical GAs have been also reported [7]. Evolutionary methods have also been used successfully to study classical cryptography [8, 9]. Only recently, they have been applied to the study of quantum cryptography [10, 11].

In [11], a genetic algorithm (GA) was proposed to optimize QKD protocols for specific input channels (representing, for example, a particular attack being launched against a system). However, the approach in [11] required the user to provide a fixed template specifying an abstract protocol from which the GA would optimize certain user-specified parameters. Thus, the GA was not free to explore truly innovative approaches - instead, it was forced to search for protocols conforming to this predetermined template. Furthermore, this template needed to be constructed by the user before use.

In this work, we reconsider this problem and use an evolutionary algorithm (more precisely, evolution strategy) to discover optimized QKD protocols, designed to efficiently operate over a given, observed, quantum channel. Unlike prior work, our system will not be forced to use a user-defined template. Instead, our approach will simply be given a quantum communication channel (without explicit rules on how to access it) and must evolve a protocol out of *quantum circuits* allowing our system, in theory, to produce arbitrarily complex quantum communication protocols. While authors have considered using evolutionary algorithms to construct quantum circuits for *algorithms* [12, 13], we are the first, to our knowledge, to apply these techniques to the construction of optimized quantum cryptographic protocols using state-of-the-art definitions of security in that field.

There are several advantages to this approach. First, it allows researchers to investigate over what channels QKD is even possible. While theoretical upper-bounds are known, it is not known whether these are tight [14]. Our system may aid researchers in this investigation. Secondly, and more practically, one may eventually envision a future quantum communication infrastructure whereby users have access to adjustable communication equipment. Users A and B may then, on start-up (or intermittently during operation), run a standard quantum tomographic protocol to estimate the channel noise (producing the current "noise signature"), provide these measurement statistics to our algorithm which will then produce a QKD protocol optimized to counter the observed channel noise and maximize efficiency. Users may then configure their quantum communication equipment to run this protocol. This process can be repeated periodically to account for changes in operating conditions (e.g., changes in attack strategy or environmental conditions).

Our system, as we will demonstrate, is able to produce QKD protocols with a higher communication rate than standard state-of-the-art protocols are capable of producing over certain channels. *Indeed, our system can even find protocols where standard protocols would fail.* Our system is also easier to use than prior work in [11] and, since we are evolving quantum circuits, the protocols output by our system may be easier to implement in practice than those obtained in [11]. *Thus, our approach has the potential to greatly increase the efficiency of a future quantum communication network.* As quantum communication is a viable technology now, the methodology we are developing may have the potential to create a more efficient, and robust, secure communication infrastructure.

2 Quantum Communication and Key Distribution

In this section, we introduce some general quantum communication concepts and notation. For more information on this subject, the reader is referred to [15].

A classical bit exists in one of two states 0 or 1; the state of a classical bit can always be determined with certainty and classical bits may be copied arbitrarily. A *quantum bit* or *qubit*, however, can be prepared in infinitely many possible states. More precisely, a qubit is modeled mathematically as a normalized vector in $\mathbb{C}^2$. Thus, any arbitrary (normalized) vector in this space represents a possible qubit state. Furthermore, "reading" a qubit (called *measuring*) is a probabilistic process which potentially destroys the original state; finally, a qubit cannot be copied without potentially destroying it.

An arbitrary qubit is denoted by $|\psi\rangle \in \mathbb{C}^2$. Let $\{|0\rangle, |1\rangle\}$ be an orthonormal basis in which case we may write $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Normalization requires $|\alpha|^2 + |\beta|^2 = 1$. The process of measuring a qubit involves first picking an orthonormal basis and then writing $|\psi\rangle$ as a linear combination of these basis vectors (called a *superposition*). Following this, the actual measurement apparatus will take as input the given quantum state, and output one of the two basis states. The probability of observing a particular basis state is simply the norm squared of the coefficient in front of the basis vector. For example, measuring $|\psi\rangle$ in the $\{|0\rangle, |1\rangle\}$ basis produces an output of $|0\rangle$ with probability $|\alpha|^2$; otherwise the output is $|1\rangle$ with probability $|\beta|^2$. Note that, once a qubit has been measured, it collapses to the observed outcome. Thus, not only are measurements probabilistic, but they also *disturb* the original state, projecting it to the observed basis vector. These measurement operations are irreversible.

Besides the $\{|0\rangle, |1\rangle\}$ basis (called the Z basis), two other important bases are the $X = \{|+\rangle, |-\rangle\}$ and $Y = \{|0_Y\rangle, |1_Y\rangle\}$ basis. These states are defined: $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$ and $|j_Y\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i(-1)^j|1\rangle)$.

Qubits are two dimensional systems; more generally, we may model an n-dimensional quantum state as an element in a Hilbert space of dimension n. Since such a space is isomorphic to $\mathbb{C}^n$, an n-dimensional state $|\psi\rangle$ is simply a normalized vector in $\mathbb{C}^n$; i.e., $|\psi\rangle = (\alpha_1, \cdots, \alpha_n)^T$ (transposed as we view these as column vectors). We denote by $\langle\psi|$ to be the conjugate transpose of $|\psi\rangle$. Note that $\langle\phi| \cdot |\psi\rangle$ is simply the inner-product of these two vectors. Since this is such an important operation, the notation is simplified to $\langle\phi|\psi\rangle$.

Given two quantum states $|\psi\rangle \in \mathbb{C}^n$ and $|\phi\rangle \in \mathbb{C}^m$, we model the *joint state* as the tensor product $|\psi\rangle \otimes |\phi\rangle \in \mathbb{C}^n \otimes \mathbb{C}^m \cong \mathbb{C}^{nm}$. As vectors, if $|\psi\rangle = (\alpha_1, \cdots, \alpha_n)^T$, then $|\psi\rangle \otimes |\phi\rangle = (\alpha_1|\phi\rangle, \cdots, \alpha_n|\phi\rangle)^T$. To simplify notation we often write $|\psi\rangle|\phi\rangle$ or even $|\psi, \phi\rangle$.

While measurements irreversibly cause the quantum state to collapse to the observed basis vector, a second operation allowed by the laws of quantum physics is state evolution via a unitary operator. U is unitary is $UU^* = U^*U = I$ (where we write U^* to mean the conjugate transpose of U). Since we are in the finite dimensional setting, one may view U as an $n \times n$ matrix satisfying this required condition. Given an input state $|\psi\rangle \in \mathbb{C}^n$, the state after evolution via U is modeled simply as the result of the matrix multiplication $U|\psi\rangle$. If U and V are

both unitary, then $U \otimes V$ is also a unitary operator acting on the tensor space with its action defined as: $(U \otimes V)\,|\psi\rangle \otimes |\phi\rangle = U\,|\psi\rangle \otimes V\,|\phi\rangle$.

Given a statistical ensemble of states $|\psi_i\rangle$ prepared with probability p_i, we may model this as a *density operator* $\rho = \sum_i p_i\,|\psi_i\rangle\,\langle\psi_i|$. Such a state may arise after a measurement is made (since a measurement is a probabilistic process causing the state to collapse to different vectors $|\psi_i\rangle$ with probabilities p_i). More generally, a density operator is a Hermitian positive semi-definite operator of a unit trace.

We may perform various important information theoretic computations on density operators. If ρ_{AE} is a density operator acting on $\mathbb{C}^n \otimes \mathbb{C}^m$, then we write $S(AE)_\rho$ to mean the *von Neumann entropy* of the operator ρ_{AE}. For finite dimensional systems, this is simply: $S(AE)_\rho = -\sum_i \lambda_i \log_2 \lambda_i$, where $\{\lambda_i\}$ are the eigenvalues of ρ_{AE}. The conditional von Neumann entropy is denoted $S(A|E)_\rho = S(AE)_\rho - S(E)_\rho$, where $S(E)_\rho$ is computed using the eigenvalues of ρ_E, where $\rho_E = tr_A \rho_{AE}$ i.e., ρ_E is the result of "tracing out" the A system. If we write $\rho_{AE} = \sum_{i,j} |i\rangle\,\langle j| \otimes \rho_E^{(i,j)}$, then $\rho_E = \sum_i \rho_E^{(i,i)}$.

2.1 Quantum Key Distribution

A QKD protocol's goal is to establish a shared secret key between two parties A and B, secure against an all-powerful adversary E. To achieve this, A and B are allowed to use a quantum communication channel permitting qubits to be sent between each. Furthermore, a classical authenticated channel is given, which allows A and B to send messages in an authenticated, *but not secret* way. That is, an adversary E may read anything sent on this authenticated channel, but may not write to it. The adversary is allowed to perform any attack on the quantum channel (as allowed by quantum physics).

Such protocols consist of two distinct stages: first a *quantum communication stage* which consists of numerous iterations, each treated independently and identically, whereby A sends qubits to B in a variety of ways according to the rules specified by the protocol; B receives these qubits, performs some measurement on them, and interprets the measurement result. A single iteration can yield at most one raw key-bit (sometimes an iteration is discarded, for example, B's measurement outcome may be "inconclusive" as determined by the protocol). Ultimately, the goal of this stage is to output a *raw key* which is a classical string, N bits long, that is partially correlated, and partially secret. The second stage of a QKD protocol, *information reconciliation* performs an error-correcting protocol (done over the authenticated channel thus leaking more information to E essentially "for free") followed by a privacy amplification protocol. The end result is a $\ell(N)$-bit secret key which may be used for other cryptographic purposes.

We consider *collective* attacks where E treats each iteration of the quantum communication stage independently and identically. Usually this is sufficient to prove security against arbitrary general attacks [16]. Such attacks may be modeled as a unitary operator U acting on the qubit and E's private quantum memory (modeled as a vector space $\mathbb{C}^n$). Without loss of generality, we may

assume E's memory is initially cleared to some "zero" state $|0\rangle_E \in \mathcal{H}_E$ and so write U's action as follows:

$$U|0,0\rangle_{TE} = |0,e_0\rangle + |1,e_1\rangle \qquad U|1,0\rangle_{TE} = |0,e_2\rangle + |1,e_3\rangle\,, \tag{1}$$

where the states $|e_i\rangle$ are arbitrary elements in $\mathbb{C}^n$ (though unitarity of U imposes important restrictions on them that we will take advantage of later). Due to linearity, this above definition is enough to completely define E's attack on any arbitrary qubit A may send.

Given a protocol and a description of the attack, one may describe a single iteration of the protocol as a density operator ρ_{ABE}. Then, the Devetak-Winter key-rate equation applies [17, 3]:

$$r(U) = \lim_{N\to\infty} \frac{\ell(N)}{N} = S(A|E)_\rho - H(A|B), \tag{2}$$

where $S(A|E)$ is the von Neumann entropy discussed earlier, and $H(A|B)$ is the conditional Shannon entropy. Of course, we must assume the worst case in that E chooses an optimal attack U. However, due to the nature of quantum communication, different types of attacks have, in a way, different "noise signatures" and, so, A and B can determine a set Γ_ν, where ν is a list of certain important measurable noise statistics in the channel. Thus, while it is not known for certain what attack was used, it can be guaranteed that the attack $U \in \Gamma_\nu$. Therefore, the actual key-rate is: $r(\nu) = \inf_{U\in\Gamma_\nu} r(U)$.

It was shown in [11, 18] how to construct Γ_ν, to arbitrary levels of precision, in order to compute r. The noise signature ν is constructed from a standard quantum tomography protocol that users may run before using our algorithm. In particular, this signature consists of various probabilities $p_{i,j}$ which denotes the probability that B observes $|j\rangle$ if A sends $|i\rangle$ (and conditioning on A and B choosing the correct basis for such an outcome to occur), where $i \in \{0, 1, +, 0_Y\}$ and $j \in \{0, 1, +, -, 0_Y, 1_Y\}$ (note the asymmetry in the sending set versus the receiving set is intentional).

From this signature, a straight-forward process exists to construct a set $\widetilde{\Gamma}_\nu$ consisting of tuples of the form $(|e_0\rangle, \cdots, |e_3\rangle)$ such that the following properties are satisfied:

1: For every $(|e_0\rangle, \cdots, |e_3\rangle) \in \widetilde{\Gamma}_\nu$, there exists a unitary operator $U \in \Gamma_\nu$ that agrees with Eq. 1. That is, the attack is unitary (so could be implemented in theory) and it agrees with the observed noise signature ν.

2: For every $U \in \Gamma_\nu$, there exists $(|e_0\rangle, \cdots, |e_3\rangle) \in \widetilde{\Gamma}_\nu$ such that the key-rate if E used attack U is equal to the key-rate produced by the attack described by vectors $(|e_0\rangle, \cdots, |e_3\rangle)$ up to an arbitrary, user-defined, level of precision. That is, this construction does not "miss" any important attacks which minimize the key-rate.

For more information on this process of constructing $\widetilde{\Gamma}_\nu$, the reader is referred to [11, 18] (in particular Algorithm 1 from [11]).

2.2 Envisioned System

As stated, all attacks by an adversary induce a particular "noise signature" and, ordinarily, QKD protocols are constructed and then analyzed to see which quantum attacks (i.e., what noise signatures) it is secure against. If an attacker is performing an attack with a noise level outside the known acceptable limits of the protocol being implemented (or if, even, just natural noise is inducing this noise), parties must simply abort, or try an alternative protocol and hope it too can at least operate over the channel. Even if a protocol is secure against this attack, however, it may be inefficient, requiring the transmission of thousands of qubits for one single secure key bit.

We are proposing, instead, to produce protocols optimized to counteract a specific noise signature as observed by the users. We envision users A and B having access to standard quantum technology, capable of sending and receiving qubits. Users will begin, after connecting their devices to the quantum channel, by performing a standard quantum tomography protocol, whereby A sends, randomly, qubits prepared in the X, Y, or Z bases and with B measuring, independently of A, in a random basis. Users then use the authenticated classical channel to disclose all measurement results, thus allowing them to determine the noise signature ν.

One of the users (either A or B) will then run our algorithm. The algorithm will take in this noise signature, and, through the use of a genetic algorithm, produce a QKD protocol as a circuit consisting of rudimentary gates. These gates may be specified by A and B - that is, they represent basic, low-level quantum operations which the users' hardware can actually support. With current technology, this gate-set would be limited; however, our system is flexible enough that, should in the future more complicated gates be implemented, the users may simply insert a description of these gates (as unitary matrices) into our algorithm and it will automatically incorporate them in new protocol generations.

After running our system, users are provided with a complete optimized protocol. The user running our algorithm (and thus who holds the description of the protocol) will send the protocol description to the other user, thus allowing both parties to configure their equipment properly. This transmission is done over the authenticated channel so that the adversary cannot tamper with the description to her benefit. The adversary can, however, learn the protocol in its entirety (thus, the protocol itself is never actually secret).

There are two things that E can do to take advantage of this knowledge of the evolved protocol. She can change her old attack (used during the quantum tomography protocol) to a new one such that the noise signature remains the same. Alternatively, she can change her attack to a new attack with a different noise signature. Our algorithm's security analysis will ensure that the protocol evolved is secure against *any* attack with the same noise signature (thus, eliminating the first threat). To ensure security against the second threat (E changing her attack to one with a different noise signature), users must periodically, and randomly (without warning to E), re-run the quantum tomography protocol to ensure that the noise signature did not change.

If the noise levels do change, our algorithm may simply be re-run to produce a new protocol to counter-act this new attack. In practice, one may consider running this system in large blocks of iterations; each block consisting of "real" iterations (i.e., iterations where the constructed protocol was executed) and "test" iterations (those used only for verifying the noise signature did not change). If, after the execution of a large block, the noise signature has changed drastically (small changes can be handled easily due to the continuity of von Neumann entropy) users must discard this block, re-run the algorithm, and try again. The reader may be concerned that this allows E to easily create a denial-of-service attack (where users are constantly discarding and trying again) - however there are easier ways for E to create such denial of service attacks against *any* QKD protocol and so this threat is not unique to our system, but common throughout any QKD protocol. In this work, we do not consider changes to the noise signature (and simply assume E keeps her attack - or rather the noise it induces - constant). There may be very interesting future work directions in discovering a potentially better method of dealing with them.

3 Our Algorithm

In our work, we will be evolving protocols modeled as *quantum circuits*. A quantum circuit operates over m *wires*, each wire "carrying" a qubit. Thus, the joint state modeling a system running on m wires (i.e., m qubits) is $\mathbb{C}^{2^m}$. On each wire, a *gate* may be placed, which are simply unitary matrices acting on the qubit wire. Common gates include:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{3}$$

$$R(p, \theta, \psi) = \begin{pmatrix} \sqrt{p}e^{i\theta} & \sqrt{1-p}e^{-i\theta} \\ \sqrt{1-p}e^{i\psi} & -\sqrt{p}e^{-i\psi} \end{pmatrix}$$

Gates may also be applied in a "control" mode in which case they act on two wires: a *target wire* and a *control wire*. In this mode, the gate will only act on the target wire if the control wire is in a $|1\rangle = (0,1)^T$ state. This operation may be done in a unitary manner. Finally, besides unitary gates, a measurement may be performed on the wire collapsing it, in a probabilistic manner, to a classical "0" or "1" state.

A protocol is, essentially, a probabilistic computation performed by parties. Any classical or quantum computation may be performed on a quantum circuit. Therefore, we will evolve protocols as quantum circuits - one for A and one for B. Unlike prior work in [11], where protocols are evolved based on solutions to free parameters within a confined template, this new mechanism will allow the EA to discover new solutions not restricted to a given template.

Circuits, as mentioned, operate over several wires. By measuring a wire, it becomes a classical wire (only modeled as a quantum basis state). Of great importance to any QKD protocol are the following:

1. A single wire to carry a qubit from A to B (passing through the adversary E).
2. Each party A and B must have, at the end of the protocol, a classical wire (i.e., a quantum wire that was measured to produce a classical output). This wire will store their key-bit for the iteration.
3. Each party may have access to additional "optional" wires to be used arbitrarily.

A diagram of this scenario is shown in Figure 1. Note that we do not need to provide additional randomness to our method – indeed, if a protocol requires randomness, it can first apply an $R(p, 0, 0)$ gate and then a measurement producing a classical output of 0 with probability p and 1 with probability $1 - p$. Thus this mechanism is sufficient to model protocols involving quantum and classical computation, along with random choices.

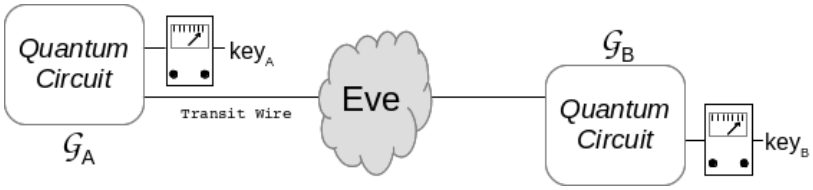

Fig. 1. A QKD protocol as two circuits $\mathcal{G}_A$ and $\mathcal{G}_B$. Circuit $\mathcal{G}_A$ is run first after which, the $\texttt{key}_A$ wire is measured yielding a classical bit. After $\mathcal{G}_A$, Eve is allowed to attack the transit line. Finally, B's circuit is run, acting on the transit wire, and additional wires private to B. Then, B's $\texttt{key}_B$ wire is measured.

3.1 Evolutionary Algorithm Approach and Parameters

A protocol is a specified process for A and B. We restrict ourselves currently to *one-way* QKD protocols whereby A sends qubits to B. In general, the qubit that A sends should depend in some manner (possibly random) on her key-bit choice for the iteration while B's measurement result should lead (again with some potential randomized post-processing) to his key-bit (which should be correlated to A's choice of key-bit). This process then repeats in i.i.d. way over subsequent iterations yielding a raw-key. As discussed earlier, the process for A and B will be described as a quantum circuit.

In particular, a protocol is a pair $\Pi = (\mathcal{G}_A, \mathcal{G}_B)$ where $\mathcal{G}_A = (g_{A,1}, \cdots, g_{A,n_A})$ is a list of gates (i.e., a quantum circuit) which A applies in sequence to her wires (similarly for $\mathcal{G}_B$ which is B's half of the protocol). Gates in $\mathcal{G}_A$ can only be applied to the transit wire and those wires private to A; similarly, gates in $\mathcal{G}_B$ may be applied only to the transit wire and those wires private to B. A gate, abstractly, simply specifies a type (we support H, X, $R(p, \theta, \psi)$, and a measurement operation, however, other gates or operations may be added or removed easily) and what wire it is applied to. Any gate may be added in a

"control" mode, thus there is an additional "control" flag which, if true, will cause the gate to only be applied if a specified target wire is in a $|1\rangle$ state; note this target wire is also part of the gate structure.

Translating the above said into a data structure, we encode a potential solution (a potential circuit) as a combination of different data types. More specifically, we use:

1. an integer vector to encode the gate type with values 0 to 3 (see Eq. (3) for a description of gate types with index 3 meaning a measurement operation),
2. an integer vector to encode the gate target (wire),
3. a bit string vector to denote whether a gate is in control mode (value 1) or not (value 0),
4. an integer vector to encode the gate control (wire),
5. a vector of floating-point values to encode the parameters of the R gate: p, θ, ψ. Note, if a different gate type is selected, these values are not used.

The number of elements in each vector is equal to the total number of gates in a circuit, where the first subset of gates is reserved for the A side and the remaining ones for the B side. In the evolutionary computation language, the above data structure represents an individual's chromosome; an evolutionary algorithm will keep a set of these individuals, a population, and perform the search for better individuals using various modification operators and selection methods.

The fitness of a candidate solution $\Pi = (\mathcal{G}_A, \mathcal{G}_B)$ will be its key-rate (Eq. (2)) against any attack with the given noise signature ν. To compute Eq. (2), we must not only simulate the quantum system but we must also simulate all possible $U \in \Gamma_\nu$ (or, rather, all $U \in \widetilde{\Gamma}_\nu$ which, as discussed, is sufficient to verify security in the worst-case). To do so, we use a quantum simulator which was specifically designed to work with the combination of quantum cryptography and evolutionary computation, developed originally in [10]. This simulator models arbitrary multi-user quantum states as density operators stored internally as linked-lists of so-called `KetBra` data structures.

A single `KetBra` encodes a quantity of the form: $p\,|i_1, \cdots, i_n\rangle\,\langle j_1, \cdots, j_n|$, with $p \in \mathbb{C}$ and i_k, j_k integer indices ranging from 0 to the dimension of the k'th subspace. In our case, since we are modeling circuits, each subspace is dimension two (i.e., a quantum wire) and so $i_k, j_k \in \{0, 1\}$ *except for the last subspace* which we assume is held by Eve and so can be higher dimensional. These integer values may represent basis states or arbitrary vectors to be substituted in later. For all but E's wire, these will be basis states (the actual choice of basis is not relevant to entropy computations). For E's wire, these integer indices will actually represent which of the four vectors $|e_i\rangle$ are to be placed there (see Eq. 1). A linked-list of these `KetBra` structures is taken to mean their sum. Since any density matrix may be written as a sum of terms of the form $p_{i,j}\,|\boldsymbol{i}\rangle\,\langle\boldsymbol{j}|$, this mechanism may be used to represent any finite-dimensional quantum system. Given actual vectors for the $|e_i\rangle$ it is a simple method to construct an actual density matrix and then perform the necessary entropy computations to evaluate the key-rate $S(A|E) - H(A|B)$ (all of which is already supported by the simulator).

Each wire in the protocol is indexed; for our simulation, we order the wires so that 0 through w_A (inclusive, where w_A is specified by the user) are private to A; furthermore, we enforce the condition that 0 always be considered her key-bit wire (denoted $\texttt{key}_A = 0$). Wire $T = w_A + 1$ is the "transit" wire which carries a qubit from A to B. This is simulated simply by allowing A to access this wire, followed by the adversary, followed, finally by B. Thus all parties can access wire T, but only in the prescribed order. This ordering is enforced by our fitness calculation function; indeed, if ever a candidate solution were presented for fitness evaluation which allowed A access to B's wires (or B access to A's wires), the fitness is defined to be 0 - i.e., "abort." Wires $T + 1$ through w_B (inclusive) are private to B with wire $\texttt{key}_B = T + 1$ being his key-bit wire. Finally, subspace $w_B + 1$ is Eve's private quantum memory used during her attack. We do not assume this is a wire, but a higher dimensional subspace.

Our fitness computation for Π begins by resetting the simulator to the "zero" state $1 \cdot |0, \cdots, 0\rangle \langle 0, \cdots, 0|$ (i.e., all wires and E's ancilla begin in a $|0\rangle$ state). Next, all gates in $\mathcal{G}_A$ are applied in sequence (simulating A's protocol). E then attacks (which is abstractly simulated using notation from Eq. (1). Then, the gates in $\mathcal{G}_B$ are applied in order. Finally, we force a measurement of the $\texttt{key}_A$ and $\texttt{key}_B$ wires so that they yield classical outcomes 0 or 1 (with various probabilities). At the conclusion, we have in our simulator a density operator description of our protocol stored as a linked-list of `KetBra` structures.

At this point, we have a linked-list of `KetBra` structures representing the density operator of the protocol. Using an algorithm developed in [11], we may enumerate through all potential attack vectors $|e_i\rangle$, substituting them into the density operator, and thus computing its entropy $S(A|E)$ (a simple function of the eigenvalues of the resulting matrix). We take the minimum over all possible $|e_i\rangle$ produced by the algorithm as we must assume the worst case that E chooses an optimal attack. Of course, the goal of our EA is to **maximize** this value over all circuits (protocols).

Evolutionary Algorithm: In our experiments, we use evolution strategy (ES) of the type $(\mu + \lambda)$-ES. In this algorithm, in each generation, parents compete with offspring and from their joint set, μ fittest individuals are kept. In our experiments, offspring population size λ has a value equal to 4, while the parent population size μ equals 1. For further information on ES, we refer interested readers to [19].

In the ES, the offspring may be generated using either a single parent with the mutation operator, which is the most widely used variant. Additionally, the offspring may be created by using two or more parents, which corresponds to the crossover operator. In our experiments with ES we use the mutation operator only, which takes a parent and randomly modifies a part of its genotype. In each mutation operation, first a part of the individual's data structure is selected at random; then, depending on the type of the selected part, the mutation is performed in the following manner:

1. for an integer vector, a single random element in the vector is changed to a new random value (corresponding to either another wire or another gate type being selected);
2. for a bit string vector, a single element is inverted (changing the control nature of the gate);
3. for a vector of floating-point values, a single random element is changed according to the Gaussian distribution with mean 0 and standard deviation of 1 (thus modifying the parameters of the R gate).

Consequently, for each parent individual, four new individuals are created in this way; the best of the five individuals is then selected as the new parent and the process is repeated. In all the experiments, the number of runs for each configuration is 30 and the stopping criterion is either 100 000 evaluations or maximum running time of 10 hours per run.

Apart from ES, we also experimented with a genetic algorithm, but we found ES to converge much faster. This is in part due to the simulator: the duration of a single evaluation is not constant and varies greatly depending on the solution quality. The GA tended to generate solutions which take much more time to evaluate, and this would in turn drastically slow down the convergence. The ES, on the other hand, managed to perform many more evaluations in the same amount of time, and consequently reach much better solutions on average.

Summary To summarize our approach, users begin by using their quantum equipment to run a standard quantum tomographic protocol, resulting in a noise signature ν (in our evaluations this is simulated). From this, one of the parties, either A or B, will run the algorithm, providing as input ν. Our algorithm will produce an optimized protocol, in the form of a quantum circuit consisting of gates which may be *user specified based on the capabilities of their devices*. Whichever party runs the EA will broadcast, through the authenticated public channel, the gate description so that both parties are able to configure their equipment appropriately. Note that this also gives E information on the protocol (i.e., the protocol description is not secret information once it is in operation). This is not an issue for security, so long as the noise in the channel does not alter (as our algorithm builds a protocol based on the optimal attack within Γ_ν). To enforce that E does not change the attack to one outside of Γ_ν, A and B must periodically, and randomly, re-run the tomographic protocol; should the noise signature change, they must simply re-run our EA to produce a new protocol for the new noise signature. Note that we did not consider imperfect parameter estimation, that addition would not be difficult to introduce by increasing the size of Γ_ν based on imperfect ν; we leave this as future work. A schematic diagram of our algorithm and this process is shown in Figure 2.

4 Experimental Results

We evaluate our algorithm over symmetric channels and arbitrary, asymmetric ones. A symmetric channel is parameterized by a single noise value Q (with

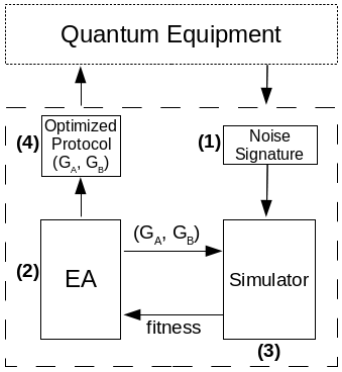

Fig. 2. A diagram of our algorithm and approach. Users begin by providing our system with the channel noise signature ν (1). The EA will evolve candidate solutions, which are pairs of circuits $(\mathcal{G}_A, \mathcal{G}_B)$ (2). Each candidate solution is sent to the simulator (3) for fitness evaluation which requires the noise signature to compute. Finally, an optimized protocol is output (4) from which the users may configure their devices to optimize the secure key distribution rate.

$Q = 0$ meaning there is no noise in the channel). In [14], it was shown that the BB84 protocol [2] (which is also generally the protocol used in practice in current-day QKD implementations) cannot be surpassed over such symmetric channels. Thus, this case serves as a useful test to verify the correctness of our algorithm (our system should be able to find a protocol with a key-rate equal to that of BB84). Table 1 shows that our approach does, indeed, find protocols that achieve the optimal BB84 rate.

In the experiments presented here, we consider scenarios where there are 5 gates and 5 wires. Two of the wires belong to the A side, two belong to the B side, and 1 wire is a joint one. Finally, out of the 5 gates, we consider a scenario where three gates belong to the A side and two gates belong to the B side. This number of gates is sufficient to construct most standard, state-of-the-art QKD protocols known today (including BB84). One could think that such a limited setting actually gives a small search space size and there is no need for evolutionary algorithms approach. However, even if we do not consider the floating-point encoding in our chromosome, we are still left with a search space size of $4^5 \times 5^5 \times 2^5 \times 3^5$ solutions, which equals to 24 883 200 000 possible configurations. Additionally, the computational bottleneck is not on the evolutionary side but on the evaluation side due to the quantum simulator complexity, which prohibits any possibility of running an exhaustive search.

All the experiments suggest that the number of 100 000 evaluations is sufficient for the algorithm convergence, after which it is more efficient to restart the run. Convergence for a typical experiment conducted in 30 runs is shown in Figure 3; at each point up to the maximum number of evaluations, current best fitness values across all the runs in the form of their minimum, maximum, median and mean value are shown.

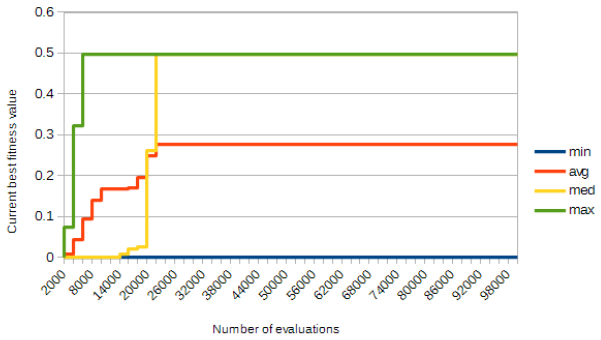

Fig. 3. Convergence rate showing best fitness value over multiple runs.

Table 1. Symmetric channel results for various levels of noise, also comparing with the BB84 protocol (the six-state version) which is known to be optimal on these channels. Our algorithm was able to evolve a protocol matching the optimal BB84 rate.

Noise	Max	Min	Avg	Std dev	BB84 (Opt.)
0.01	**0.864**	0.864	0.864	0	0.864
0.05	**0.497**	$1.45 \cdot 10^{-15}$	0.263	0.2556	0.497
0.1	**0.152**	$1.22 \cdot 10^{-15}$	0.061	0.0766	0.152
0.12	**0.035**	$3.22 \cdot 10^{-15}$	0.007	0.0152	0.035

We also test over arbitrary, asymmetric channels. For such channels, BB84 is not necessarily optimal, but no known theoretical result exists claiming how an optimal protocol should be constructed. Thus, evaluating over asymmetric channels serves as an interesting evaluation test for our system. In particular, we evaluate over the two asymmetric channels that were considered in [11] for comparison purposes. These channels are the result of an attack which could be launched against an actual QKD system; they also cause BB84 to fail (i.e., abort). Since these channels were also the ones considered in [11], we can also compare how our gate-based approach compares with the template-based version. The results of this test are summarized in Table 2. Figure 4 shows a sample protocol output by our algorithm.

5 Closing Remarks

In this paper, we showed how evolution strategy can be used to evolve quantum cryptographic protocols modeled as quantum circuits. Our approach was able to find a protocol matching the optimal BB84 key-rate for symmetric channels. We were also able to find new QKD protocols which can operate over quantum channels (i.e., against attacks) where ordinary, state-of-the-art QKD protocols would fail. In a future quantum network infrastructure, our approach would

Table 2. Asymmetric channel results. Here we test on the two randomly generated channels (i.e., attacks) considered in [11]. Note that for both of these channels, BB84 would fail to establish a key; however, our approach can find protocols with a positive key-rate. While the key-rate is not as high as the template-based approach from [11] for these two channels, this is to be expected as we are optimizing protocols built of a limited set of quantum gates. While our new system proposed here may not achieve as high a key-rate as the template-based approach, the protocols evolved here are potentially easier to implement in practice as they are based on simple gates.

Description	Max	Min	Avg	Std dev	BB84/Ref. [11]
Channel 1	**0.066**	$2.22 \cdot 10^{-16}$	$5.73 \cdot 10^{-4}$	0.00603	0(abort)/.094
Channel 2	**0.018**	$3.33 \cdot 10^{-16}$	$2.58 \cdot 10^{-4}$	0.00197	0(abort)/.042

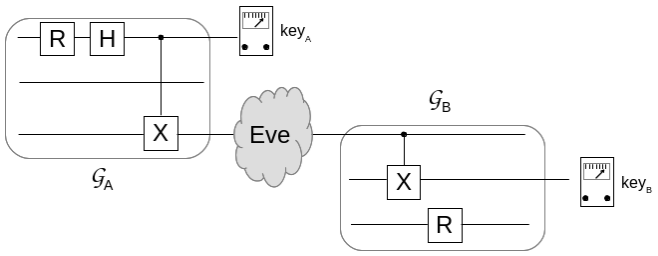

Fig. 4. A protocol output by our algorithm optimized to run on Channel 1.

allow users to easily adapt their communication protocols to counter quantum adversaries, thereby improving the efficiency of the secret communication.

Many very interesting open problems remain. It would be very interesting to provide our system with increased quantum communication resources (such as two-way channels for example) in order to improve the key-rate. Also providing our system with classical communication resources allowing for the evolution of more complex post-processing strategies, known to be important for improving the key-rate of protocols over noisy channels [3]. We suspect this is one area where our new approach of evolving quantum circuits for QKD protocols would be greatly beneficial (prior work in this area would require users to write out in detail an abstract template which would be challenging for these more powerful quantum and classical resources). It would also be very interesting to take into account practical imperfections in the optical devices used. Other areas of future work could include having the number of gates and wires as part of the solution, however, this would require improvements in the computational speed of the simulator used.

References

1. Valerio Scarani, Helle Bechmann-Pasquinucci, Nicolas J. Cerf, Miloslav Dušek, Norbert Lütkenhaus, and Momtchil Peev. The security of practical quantum key distribution. *Rev. Mod. Phys.*, 81:1301–1350, Sep 2009.

2. 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.
3. Renato Renner, Nicolas Gisin, and Barbara Kraus. Information-theoretic security proof for quantum-key-distribution protocols. *Phys. Rev. A*, 72:012332, Jul 2005.
4. L. Spector. *Automatic Quantum Computer Programming: A Genetic Programming Approach*. Kluwer Academic Publishers, Boston, MA, 2004.
5. Xiaoxiao Wang, Licheng Jiao, Yangyang Li, Yutao Qi, and Jianshe Wu. A variable-length chromosome evolutionary algorithm for reversible circuit synthesis. *Journal of Multiple-Valued Logic & Soft Computing*, 25(6), 2015.
6. Mustapha Y Abubakar, Low Tang Jung, Nordin Zakaria, Ahmed Younes, and Abdel-Haleem Abdel-Aty. Reversible circuit synthesis by genetic programming using dynamic gate libraries. *Quantum Information Processing*, 16(6):160, 2017.
7. Bart Rylander, Terry Soule, James Foster, and Jim Alves-Foss. Quantum evolutionary programming. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, pages 1005–1011. Morgan Kaufmann Publishers Inc., 2001.
8. Stjepan Picek and Marin Golub. On evolutionary computation methods in cryptography. In *MIPRO, 2011 Proc. 34th International Convention*, pages 1496–1501. IEEE, 2011.
9. Stjepan Picek, Luca Mariot, Alberto Leporati, and Domagoj Jakobovic. Evolving s-boxes based on cellular automata with genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 251–252. ACM, 2017.
10. Walter O Krawec. A genetic algorithm to analyze the security of quantum cryptographic protocols. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 2098–2105. IEEE, 2016.
11. Walter O Krawec, Michael G Nelson, and Eric P Geiss. Automatic generation of optimal quantum key distribution protocols. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1153–1160. ACM, 2017.
12. Ben IP Rubinstein. Evolving quantum circuits using genetic programming. In *Proc. 2001 Congress on Evolutionary Computation*, pages 144–151. IEEE, 2001.
13. Lee Spector and Jon Klein. Machine invention of quantum computing circuits by means of genetic programming. *AI EDAM*, 22(3):275–283, 2008.
14. Joonwoo Bae and Antonio Acín. Key distillation from quantum channels using two-way communication protocols. *Physical Review A*, 75(1):012334, 2007.
15. M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, MA, 2000.
16. Matthias Christandl, Robert Konig, and Renato Renner. Postselection technique for quantum channels with applications to quantum cryptography. *Phys. Rev. Lett.*, 102:020504, Jan 2009.
17. Igor Devetak and Andreas Winter. Distillation of secret key and entanglement from quantum states. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 461(2053):207–235, 2005.
18. Walter O. Krawec. Quantum key distribution with mismatched measurements over arbitrary channels. *Quantum Information and Computation*, 17(3):209–241, 2017.
19. T Bäck, D.B Fogel, and Z Michalewicz, editors. *Evolutionary Computation 1: Basic Algorithms and Operators*. Institute of Physics Publishing, Bristol, 2000.