Chapter 5

Strengthening Symmetric Keys with Quantum Information

5.1 Introduction

In this chapter, we turn to a different topic in quantum cryptography.

The field of quantum cryptography studies cryptographic protocols that take advantage of the properties of quantum information. The protocols are typically designed so that security rests on two assumptions: (1) quantum mechanics is an accurate description of the world and (2) the quantum devices implementing the protocol are vulnerability-free (i.e. they behave according to how they are modeled in the security proof). In theory, quantum protocols can offer higher security than non-quantum[1] protocols, since they don't need to rely on unproven complexity-theoretic assumptions, e.g. that it's computationally hard to break a cipher like the Advanced Encryption Standard (AES) [DR13].

However, some caution is called for. Building quantum devices that behave as they are modelled in the security proofs seems to be hard, and many attacks on real-world implementations of quantum cryptography are known, taking advantage of flaws in the devices. Attacks that blind and manipulate single-photon detectors [LWW$^+$10] and attacks that damage components by shining a laser through the optical connection [MBC$^+$16] are just some examples. On the other hand, the complexity-theoretic assumptions of non-quantum cryptography have held up over time. For example, the AES cipher has been

[1] We avoid the term "classical cryptography" because to many cryptographers, "classical cryptography" refers to the simple ciphers used in antiquity and other pre-1970's cryptography. We use "non-quantum cryptography" to refer to the collection of modern ciphers and primitives in widespread use today.

analyzed by cryptographers for two decades since 1998, when the cryptanalysis competition that selected it for standardization began. It remains unbroken. So, it seems that quantum protocols remove complexity-theoretic assumptions that have held up well in practice and add new implementation-correctness assumptions that (at least currently) don't seem to hold up.

Of course, non-quantum cryptography needs to assume the implementation is correct, too, and many implementation problems have lead to attacks (e.g. Heartbleed [DKA+14]). However, quantum protocols are implemented by physical devices that are open to physical attacks, and non-quantum protocols are typically implemented in software on devices the attacker can't physically interact with, so the two kinds of implementation-correctness problems seem qualitatively different. Problems with non-quantum protocols can usually be fixed with a software patch, whereas insecure quantum devices might need to be physically upgraded. More research and engineering efforts will be needed to determine if the quantum devices can be made as reliable as the non-quantum ones.

If it is indeed harder to secure quantum crypto devices than non-quantum ones, it would be a mistake to switch from a non-quantum protocol to a quantum protocol that accomplishes the same task. On the other hand, if there are things that quantum protocols can do that non-quantum ones can't, then it may be worth the added risk of using quantum devices for those applications. In this chapter, we try to find a new use for quantum cryptography by asking whether *combining* quantum cryptography with complexity-theoretic assumptions will let us accomplish new cryptographic feats that are not possible using information-theoretic quantum cryptography or complexity-theoretic non-quantum cryptography on their own.

One possibility we explore is a hypothetical primitive that we call an "offline key expander", or an OKE for short. If OKEs exist, they make it possible to increase the strength of a symmetric encryption key against brute-force attacks in exchange for giving the attacker a one-time chance to defeat the scheme quickly. In contrast to protocols like BB84 [BB14] where two parties communicate over a network to lengthen a short shared secret key, OKEs work entirely offline and would be suitable for encrypting data at rest. OKEs are impossible in a non-quantum world, because as long as there is a way to check if you have the right key, the only way to make a brute-force key-guessing attack more expensive is to make it costly to check if you have the right key, and that cost will have to be paid by the legitimate key-holder every time they decrypt their data. On the other hand, quantum information cannot always be perfectly cloned, and measurement of a quantum state can disturb it, so there is hope that OKEs might be possible by hiding a longer key in a quantum state.

If secure OKEs exist, they represent an entirely new kind of cryptographic capability, and would make the effort to build hack-proof quantum-cryptographic devices more worthwhile. The primary practical application of OKEs would be to expand short easy-to-remember passwords into longer, more-secure, cryptographic keys.

Unfortunately, we aren't able to prove that offline key expanders exist, nor are we able to prove that they don't exist. In the following sections, we formally define offline key expanders, then we describe a scheme that seems like it could be a secure OKE. We prove that the scheme is actually insecure for some values of its parameters, and suggest some modifications that could make it secure in models where the adversary only has access to "weak" quantum computers.

Although our lemma showing it is possible to Grover-search a function that is approximately-implemented using only one available copy of a quantum state might find applications elsewhere, the primary purpose of this chapter is to motivate and begin exploring a new direction in quantum cryptography. Our definitions are exploratory in nature so there may be better ways of characterizing the problem we are interested in.

5.2 Offline Key Expanders

The function of an Offline Key Expander (OKE) is to enable a trade-off where the adversary is given a one-time chance at breaking an encryption scheme in exchange for making a short encryption key effectively as secure as a longer one in the case where the one-time chance attack fails. In general, an OKE will encode a long key into a quantum state in such a way that knowledge of the short key is required to decode it. In order to encrypt data with an OKE, it will need to be combined with a non-quantum encryption scheme. In the future, we would like to define OKEs so that the protected key can safely be used with *any* kind of secret-key-based non-quantum cryptography, but for now we start by studying OKEs that are used for encryption. We expect the encryption case to be similar to all the others.

A non-quantum encryption scheme is a pair $(\mathcal{E}, \mathcal{D})$ of polynomial-time classical algorithms, where $\mathcal{E} : \{0,1\}^m \times \{0,1\}^* \to \{0,1\}^*$ is a random function that takes in an m-bit key and a message of any length and encrypts it to create ciphertext, and $\mathcal{D} : \{0,1\}^m \times \{0,1\}^* \to \{0,1\}^* \cup \{\perp\}$ is the deterministic decryption function, undoing the encryption or indicating failure. For correctness, we require that $\Pr[\mathcal{D}_k(\mathcal{E}_k(M)) = M] = 1$ for all keys k and messages M. In order to be secure, $(\mathcal{E}, \mathcal{D})$ needs to satisfy some kind of strict game-based security definition, like IND-CPA or IND-CCA2 [BDPR98]. To keep our discussion simple, we will assume the following two idealized properties:

1. If an adversary is allowed to choose a message M, and is then given $\mathcal{E}_k(M)$ for a random $k \in \{0,1\}^m$, in order to have a high chance of guessing k, a quantum adversary needs to perform on the order of $2^{m/2}$ operations, corresponding Grover-searching the keyspace.

2. For large enough messages M, say $M \in \{0,1\}^{10m}$, it is likely (i.e. more than 99%) that $\mathcal{D}_{k_1}(\mathcal{E}_{k_2}(M)) = M \implies k_1 = k_2$ for all keys $k_1, k_2 \in \{0,1\}^m$.

These conditions are neither necessary nor sufficient for security. For example, if $\mathcal{E}_k(M) = M$ for all k and M, then even unbounded adversaries will have a negligible chance of guessing the key given the ciphertext, yet the encryption scheme is not secure. And a secure encryption scheme can fail the second property (while still being secure) by ignoring one bit of its key. Nevertheless, our goal is to take the first steps to answer questions about OKEs, so it makes sense to start with an idealized model. Encryption schemes that satisfy both of our properties are believed to exist, for example it would be surprising if AES-GCM [MV04] failed to satisfy either one.

Now we are prepared to formally define what an offline key expander is. An n-to-m ($n \leq m$) OKE is a pair of polynomial-time quantum algorithms (E, D). The algorithm E, on inputs $s \in \{0,1\}^m$ and $k \in \{0,1\}^n$, produces a quantum state $E(s,k) \in \mathrm{D}(\mathcal{X})$ (where $\mathcal{X}$ is some finite-dimensional Hilbert space whose size can depend on $|s|$ and $|k|$). The algorithm D takes a string $s \in \{0,1\}^m$ and state $\rho \in D(\mathcal{X})$ as input and probabilistically produces a result $D(s,\rho) \in \{0,1\}^m$. The algorithms E and D must satisfy the following correctness and security properties:

1. **Correctness.** We require that $\Pr[D(s, E(s,k)) = k] = 1$ for all possible s and k, so that decoding the long key always works reliably given the correct short key.

2. **Security.** An OKE adversary is a pair (A, M) where $M \in \{0,1\}^*$ is the adversary's chosen-plaintext and A is a quantum algorithm which is given the state $E(s,k)$ and $\mathcal{E}_k(M)$ as input for a random s and k. The algorithm must output a guess $k' \in \{0,1\}^n$ for k.

   Let $(\mathcal{E}, \mathcal{D})$ be a secure non-quantum encryption scheme for encrypting messages with m-bit keys. We say that an n-to-m OKE is (p,t)-secure for $(\mathcal{E}, \mathcal{D})$ if for all quantum adversaries A allowed to run for time t, $\Pr[A(E(s,k), \mathcal{E}_k(M)) = k] \leq p$, where M is the adversary's chosen plaintext and the probability is taken uniformly over all choices of k, s, and the adversary's own bits of randomness. An n-to-m OKE is (p,t)-secure if it is (p,t)-secure for all secure non-quantum encryptions schemes $(\mathcal{E}, \mathcal{D})$.

The idea is that E hides an n-bit key k protected by an m-bit key s in the state $E(k, s)$, and then D can be used—with knowledge of s—to recover k .

The reason we have separated the notion of an OKE instead of defining a stronger-than-normal kind of classical-plaintext-to-quantum-ciphertext encryption is that we want OKEs to be usable for more than just encryption. Furthermore, when they *are* used for encryption, we would like to be able to encrypt large amounts (i.e. terabytes) of data stored on classical hard drives; a classical-to-quantum cipher would require terabytes of *quantum* storage to do this. A classical-to-quantum cipher that's secure against more powerful adversaries than non-quantum ciphers can be (for equal key lengths) could be used to encrypt a longer key and construct an OKE, but the reverse is not necessarily true, because an OKE might require the short key to be unique for each use.

We've kept the definition simple in order to avoid adding unnecessary complexity to our analysis before we understand the basics. An improved definition that would be more practically-relevant should:

- Require the ability to successfully decode the key even after a small number of failed guesses (especially important if the short key is derived from a password).
- Provide stronger security guarantees. Under the current definition, an OKE only fails to be secure if an adversary can recover the entire key. In actual practice, we will need the OKE—combined with non-quantum encryption—to satisfy game-based definitions analogous to IND-CPA.

Our definition is good enough to begin exploring the central question of this chapter: for which values of n, m, p, and t do (p, t)-secure n-to-m OKEs exist?

We can make some trivial observations:

- A $(2^{-n}, \mathsf{poly}(n))$-secure n-to-n OKE exists: encrypt the "long" key with the "short" key using the classical one-time pad (both keys are the same length in this case).
- No $(\epsilon, \Omega(\mathsf{poly}(n)))$-secure n-to-m OKEs exist for $\epsilon < 2^{-n}$, since an adversary who runs the decoding algorithm with a random short key will get it right and recover the long key with probability at least 2^{-n}.
- No $(p, \Omega(2^n))$-secure n-to-$2n$ OKEs exist for any $p < 1$, since the adversary can ignore the quantum state and use Grover's algorithm to find the $2n$-bit key in 2^n time.

For practical applications, it would be sufficient to have a $(0.0001, 2^{80})$-secure 64-to-256 OKE, or an OKE for any other similar concrete values of p, t, n, and m. If such an OKE existed, it could be used to make an easy-to-memorize 64-bit-equivalent password as secure as an 80-bit key, aside from the one-time (at most) one-in-10,000 chance that the adversary has of breaking the scheme.

Unfortunately, other than the trivial facts listed above, we weren't able to determine whether (p,t)-secure n-to-m OKEs exist for any specific values of p, t, n, and m, nor were we able to rule out the existence of OKEs where p, t, and m are functions of n as n grows to infinity. The difficulty comes from the fact that the definition of security provides the adversary with an encrypted message of their choice. Since it is possible to recover the key from the encrypted message by brute-force guessing-and-checking, the entropy of the key relative to the adversary is zero, and we can't use purely information-theoretic tools to analyze the security of an OKE scheme. Another difficulty comes from the fact that we are trying to improve the security of an n-bit key, which means we're working in a model where the adversary is allowed to run for an exponential 2^n amount of time but not $(2^n)^k \in \mathsf{poly}(2^n)$ time ($m = kn$). The tools from complexity theory for dealing with polynomial (rather than exponential) differences in algorithm cost are not very well developed. For example in non-quantum cryptography it is usually assumed that the adversaries run in $\mathsf{poly}(n)$ time and security is proven with polynomial-time reductions, but this cannot be done for OKEs since when we are considering adversaries who can run for 2^n steps, breaking the OKE by brute-forcing the long key is technically "polynomial time."

We leave the problem of determining what kinds of OKEs exist for future work, and proceed in the next section by designing a seemingly-reasonable candidate OKE scheme. We show that some of its variants can be broken with high probability by a quantum attack in $O(2^{n/2}\mathsf{poly}(n))$ time.

5.3 PCC: A Candidate OKE

One idea for implementing an offline key expander comes from the BB84 protocol, where a long random key is transmitted in secret between two parties by applying *conjugate coding.* Conjugate coding works by encoding bits of the key randomly either in the $\{|0\rangle, |1\rangle\}$ basis or the $\{|+\rangle, |-\rangle\}$ basis. It is extremely unlikely for any adversary to be able to learn all of the key bits given just one copy of the state, since it is impossible to perfectly distinguish the states $\{|0\rangle, |+\rangle\}$ (key bit is 0) from the states $\{|1\rangle, |-\rangle\}$ (key bit is 1). In the BB84 protocol, if the adversary tries to measure information about the key as it is being sent, their measurement causes some disturbance to the state and their attack will probably be detected later on in the protocol.

In order for BB84 to be secure, it's crucial for the conjugate coding bases to be selected uniformly at random and be kept secret from the adversary until later on in the protocol. To implement an OKE, perhaps it would suffice to encode the long key with conjugate coding bases *pseudorandomly* derived from the short key. Intuitively, to make a guess at the short key, the adversary should have to perform a measurement and damage the encoded key state, making it less likely they'll be able to recover the key in a later stage of their attack. To evaluate this idea, we propose a candidate n-to-ℓn OKE construction called Pseudorandom Conjugate Coding (PCC).

Definition 5.3.1 (ℓ-PCC on n bits)**.** Let $n \in \mathbb{N}$, $\ell > 2$, and $H = |+\rangle\langle 0| + |-\rangle\langle 1|$. Let $g : \{0,1\}^n \to \{0,1\}^{\lfloor \ell n \rfloor}$ be a random oracle. Let $H_s = \bigotimes_{j=1}^{\lfloor \ell n \rfloor} H^{g_j(s)}$ for all $s \in \{0,1\}^n$. Let $|E_s^k\rangle = H_s |k\rangle$ for all $s \in \{0,1\}^n$ and $k \in \{0,1\}^{\lfloor \ell n \rfloor}$. Then the ℓ-PCC scheme on n bits is (E, D) where E is defined by,

$$E(s,k) = |E_s^k\rangle\langle E_s^k|, \tag{5.1}$$

and $D(s,\rho)$ for $\rho \in \mathrm{D}(\mathbb{C}^{2^{\lfloor \ell n \rfloor}})$ is the result of applying H_s to ρ and then measuring in the computational basis.

Both E and D can be implemented by polynomial-time algorithms, and since $H^2 = \mathbb{1}$, $\Pr[D(s, E(s,k)) = k] = 1$ for all s and k, so PCC satisfies the correctness property for an n-to-$\lfloor \ell n \rfloor$ OKE. We've defined PCC using a random oracle g, but in practice g might be replaced by a pseudorandom generator (PRG) or some other kind of function that doesn't behave randomly at all like an error-correcting code. Even if g were a PRG, we want to let the adversary run for 2^n time, which is enough to break a PRG on n bits anyway.

We now investigate the PCC scheme's security.

A Naïve Attack

Recall that the security definition for OKEs provides an attacker with a message of their choice encrypted under the key the OKE is supposedly protecting. A Naïve attack against PCC tries to recover the key as best as possible from the state alone, before touching the ciphertext. One way to do so is to treat each qubit individually as an instance of the problem of distinguishing $\{|0\rangle, |+\rangle\}$ from $\{|1\rangle, |-\rangle\}$. The optimal probability of correctly distinguishing between the two sets of states is $\cos^2(\pi/8) \approx 85\%$ by measuring in the basis which is at a $\pi/8$ angle relative to $\{|0\rangle, |1\rangle\}$.

Applying this measurement to each qubit of the state produced by 3-PCC gives a string $k' \in \{0,1\}^{3n}$ such that for each bit of k', there is an independent 85% chance it is the same as the corresponding bit of k and a 15% chance it is different. Sampling k' this way is equivalent to sampling $k' = k \oplus c$ where $c \in \{0,1\}^{3n}$ is an error syndrome where each bit of c is 1 with independent probability 15%. Knowing k', we can write $k = k' \oplus c$, and so the entropy of k given k' is the same as the entropy of c, which is $-3n \cdot 0.15 \cdot \log_2(0.15) > 1.23n$. Treating the ciphertext as a black box that only allows the adversary to check a guess at the key, this suggests that the fastest an adversary could on average recover k from k' would be a $2^{1.23n}$-time classical brute-force search or perhaps some kind of $2^{1.23n/2}$-time quantum Grover-like search, both which take longer than brute-forcing an n-bit key. However, partial knowledge of the key could aid cryptanalysis attacks on the non-quantum encryption scheme, so there may be better attacks that take advantage of specific weaknesses in the encryption scheme.

If there is no better attack, then 3-PCC effectively expands an n-bit key to have $1.23n$ bits of security. In the next section we show that there is an attack that breaks ℓ-PCC for $\ell > 7.59$ with high probability in $2^{n/2}$ quantum time. This suggests that there are better attacks against 3-PCC waiting to be found.

Breaking PCC via Approximated-Oracle Grover Search

In this section, we break ℓ-PCC for $\ell > 7.59$. Suppose, for some random $s \in \{0,1\}^n$, random $k \in \{0,1\}^{\lfloor \ell n \rfloor}$, and any M, we are given the ℓ-PCC-encoded key $\rho = E(s,k)$, and a ciphertext $C = \mathcal{E}_k(M)$. If we could somehow implement the function f defined by,

$$f(x) = \begin{cases} 1 & \text{if } x = s \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

for all $x \in \{0,1\}^n$, then we could simply Grover-search f to find s in $2^{n/2}$ time and decode k from ρ using D. But to implement f we would have to perform operations on ρ since of ρ is the only thing we have that depends on s. Any use of ρ to implement f risks damaging it and making it impossible to decode k once we know s. It turns out that we can indeed break ℓ-PCC this way, for large enough ℓ. First, we need the following lemma, which says that if we can implement a single arbitrary query to the phase-flip version f accurately enough using a pure-state ρ, and without damaging ρ too much, then a variant of Grover's algorithm can find s without doing much damage to ρ.

Lemma 5.3.1. *Let $s \in \{0,1\}^n$ and let $O_s = \sum_{x\in\{0,1\}^n} (-1)^{x=s} |x\rangle\langle x|$. Suppose U is a unitary operator and $|\psi\rangle$ is a quantum state such that for all states $|\phi\rangle \in \mathbb{C}^{2^n}$, $\| U|\phi\rangle|\psi\rangle - (O_s|\phi\rangle)\otimes|\psi\rangle\| \leq \epsilon$. Then there is a unitary operator G that can be implemented by a circuit of size $O(2^{n/2}\cdot \mathrm{size}(U))$ such that $|\langle G|0\rangle|\psi\rangle, |s\rangle|\psi\rangle\rangle| \gtrapprox 1 - 2^{-n/2} - 2^{n/2}\epsilon$.*

Proof. The operator G will be Grover's algorithm modified to use U instead of O_s. Let $s \in \{0,1\}^n$. Define $f : \{0,1\}^n \to \{0,1\}$ to be the function such that $f(x) = 1$ if and only if $x = s$. Recall how Grover's algorithm works when searching f to find s. The search starts in state $|I_0\rangle = |+\rangle^{\otimes n}$, and then the operator $R = XO_s$ is applied some $N(n) \leq 2^{n/2}$ times times in succession, generating intermediate states $|I_k\rangle = R^k|I_0\rangle$ and ending on the final state $\left|I_{N(n)}\right\rangle$ [NC11]. Here X is a unitary operator whose details are irrelevant to our proof except that $\mathrm{size}(X) \in O(n)$. Grover's algorithm promises that $|\left\langle s \middle| I_{N(n)}\right\rangle| \approx \sqrt{1-2^{-n}}$.

Now suppose we start in the state $|A_0\rangle = |I_0\rangle|\psi\rangle$ and apply the operator $(X\otimes\mathbb{1})U$ iteratively $N(n)$ times, creating the intermediate states $|A_k\rangle = ((X\otimes\mathbb{1})U)^k|A_0\rangle$ of our approximated search. Overall, our algorithm is $G = ((X\otimes\mathbb{1})U)^{N(n)}(H^{\otimes n}\otimes\mathbb{1})$. The distance between the starting states of the idealized and approximated search is $\||A_0\rangle - |I_0\rangle|\psi\rangle\| = 0$. Let $k \geq 0$. Then we can write $|A_k\rangle = |I_k\rangle|\psi\rangle + |e\rangle$ where $\||e\rangle\| = \||A_k\rangle - |I_k\rangle|\psi\rangle\|$. Now,

$$\begin{aligned}
\||A_{k+1}\rangle - |I_{k+1}\rangle|\psi\rangle\| &= \|(X\otimes\mathbb{1})U|A_k\rangle - (X\otimes\mathbb{1})(O_s\otimes\mathbb{1})|I_k\rangle|\psi\rangle\| && (5.3)\\
&= \|U|A_k\rangle - (O_s\otimes\mathbb{1})|I_k\rangle|\psi\rangle\| && (5.4)\\
&= \|U|I_k\rangle|\psi\rangle + U|e\rangle - (O_s\otimes\mathbb{1})|I_k\rangle|\psi\rangle\| && (5.5)\\
&= \|U|I_k\rangle|\psi\rangle - (O_s\otimes\mathbb{1})|I_k\rangle|\psi\rangle\| + \||e\rangle\| && (5.6)\\
&\leq \epsilon + \||A_k\rangle - |I_k\rangle|\psi\rangle\|. && (5.7)
\end{aligned}$$

So by induction, $\|\left|A_{N(n)}\right\rangle - \left|I_{N(n)}\right\rangle|\psi\rangle\| \leq N(n)\epsilon \leq 2^{n/2}\epsilon$. Now write $\left|A_{N(n)}\right\rangle = \left|I_{N(n)}\right\rangle|\psi\rangle + |e\rangle$ for some new $|e\rangle$ where $\||e\rangle\| \leq 2^{n/2}\epsilon$. We have,

$$\begin{aligned}
|\langle G|0\rangle|\psi\rangle, |s\rangle|\psi\rangle\rangle| &= |(\langle s|\langle\psi|)\left|A_{N(n)}\right\rangle| && (5.8)\\
&= |\left\langle s \middle| I_{N(n)}\right\rangle\langle\psi|\psi\rangle + (\langle s|\langle\psi|)|e\rangle| && (5.9)\\
&\geq |\langle s|I_{N(n)}\rangle| - |(\langle s|\langle\psi|)|e\rangle| && (5.10)\\
&\geq |\langle s|I_{N(n)}\rangle| - \||e\rangle\| && (5.11)\\
&\geq |\langle s|I_{N(n)}\rangle| - 2^{n/2}\epsilon && (5.12)\\
&\approx \sqrt{1-2^{-n}} - 2^{n/2}\epsilon && (5.13)\\
&\geq 1 - 2^{-n/2} - 2^{n/2}\epsilon. && (5.14)
\end{aligned}$$

It only remains to show that the circuit size of G is in $O(2^{n/2} \cdot \text{size}(U))$. G is made up of at most $2^{n/2}$ applications of X and U, plus one use of $H^{\otimes n}$ at the beginning to turn $|0\rangle$ into $|I_0\rangle = |+\rangle^{\otimes n}$. We're given that $\text{size}(X) \in O(n)$ and we can assume without loss of generality that $\text{size}(U) \in \Omega(n)$, so $\text{size}(X) \in O(\text{size}(U))$ and thus overall, $\text{size}(G) \in O(2^{n/2} \cdot \text{size}(U))$. □

We can now use this lemma to construct an attack that breaks ℓ-PCC for $\ell > 7.59$ in $O(2^{n/2} \cdot \mathsf{poly}(n))$ time.

Theorem 5.3.2. *For large enough n, there is a quantum algorithm that runs in time $O(2^{n/2} \cdot \mathsf{poly}(n))$ and with high probability recovers both the short and long key from the ℓ-PCC scheme on n bits, when $\ell > 7.59$.*

Proof. Let $\ell > 7.59$ and let (E, D) be the ℓ-PCC scheme on $n \in \mathbb{N}$ bits. Suppose we are given the state $\left|E_s^k\right\rangle = E(k, s)$ and a ciphertext $C = \mathcal{E}_k(M)$ for random unknown k and s and any message M long enough that it's likely that $\mathcal{D}_{k'}(C) = M \implies k' = k$. Then all of the following operators are unitary and can be implemented in polynomial time:

$$S = \sum_{s \in \{0,1\}^n} |s\rangle \langle s| \otimes H_s \tag{5.15}$$

$$T = \mathbb{1} \otimes \Big(\sum_{k' \in \{0,1\}^{4n}} (-1)^{k'=k} |k'\rangle \langle k'| \Big) \tag{5.16}$$

$$U = STS. \tag{5.17}$$

S is implemented in polynomial time by controlling which of $\{H_s | s \in \{0,1\}^n\}$ is applied to the second register by the value in the first register. T is implemented in polynomial time by flipping the phase if the value in the second register successfully decrypts C.

We later show that it is very likely for U and $\left|E_s^k\right\rangle$ to satisfy the preconditions for Lemma 5.3.1 with $\epsilon = 2^{1-3n/4}$. This gives us an $O(2^{n/2} \cdot \mathsf{poly}(n))$-time unitary algorithm G such that

$$|\langle G |0\rangle \left|E_s^k\right\rangle, |s\rangle \left|E_s^k\right\rangle\rangle| \gtrapprox 1 - 2^{-n/2} - 2^{n/2}\epsilon \geq 1 - 2^{2-n/4}. \tag{5.18}$$

Given G, our attack will be to apply G to the state $|0\rangle \left|E_s^k\right\rangle$ and then apply S one more time, giving us the state $|\psi\rangle = SG|0\rangle \left|E_s^k\right\rangle$. S is its own inverse, so,

$$\begin{aligned} |\langle|\psi\rangle, |s\rangle|k\rangle\rangle| &= |\langle SG|0\rangle \left|E_s^k\right\rangle, |s\rangle|k\rangle\rangle| && (5.19)\\ &= |\langle SSG|0\rangle \left|E_s^k\right\rangle, S|s\rangle|k\rangle\rangle| && (5.20)\\ &= |\langle G|0\rangle \left|E_s^k\right\rangle, |s\rangle \left|E_s^k\right\rangle\rangle| && (5.21)\\ &\gtrapprox 1 - 2^{2-n/4}. && (5.22)\end{aligned}$$

Thus, after applying G, S, and then measuring in the computational basis, we are likely to obtain k and s, breaking the security of the PCC scheme.

It remains to show that U and $\left|E_s^k\right\rangle$ satisfy the precondtions for Lemma 5.3.1 with $\epsilon = 2^{1-3n/4}$. Before we begin, it will be helpful to define $\alpha_{j,x}$ for all $j \in \{0,1\}^{\lfloor \ell n \rfloor}$ and $x \in \{0,1\}^n$ by,

$$H_x \left|E_s^k\right\rangle = \sum_{j \in \{0,1\}^{\lfloor \ell n \rfloor}} \alpha_{j,x} |j\rangle, \tag{5.23}$$

and also note that for all j and x,

$$\langle E_x^j | E_s^k \rangle = \langle j| H_x \left|E_s^k\right\rangle = \alpha_{j,x}. \tag{5.24}$$

To satisfy the lemma, we need to show that for any state $|\phi\rangle \in \mathbb{C}^{2^n}$, $\|U|\phi\rangle \left|E_s^k\right\rangle - (O_s|\phi\rangle) \left|E_s^k\right\rangle\| \le \epsilon$. The remainder of this proof will etablish an upper bound on this distance. Let $|\phi\rangle = \sum_{x \in \{0,1\}^n} \delta_x |x\rangle$ be an arbitrary state in $\mathbb{C}^{2^n}$. We will compute $U|\phi\rangle \left|E_s^k\right\rangle$ in steps.

First, applying S gives,

$$\sum_x \delta_x |x\rangle H_x \left|E_s^k\right\rangle = \sum_x \delta_x |x\rangle \sum_j \alpha_{j,x} |j\rangle. \tag{5.25}$$

Next, applying T gives,

$$\begin{aligned} \sum_x \delta_x |x\rangle \sum_j \alpha_{j,x} (-1)^{j=k} |j\rangle &= \sum_x \delta_x |x\rangle \Big(\sum_j \alpha_{j,x} |j\rangle - 2\alpha_{k,x} |k\rangle \Big) && (5.26)\\ &= \sum_x \delta_x |x\rangle \Big(H_x \left|E_s^k\right\rangle - 2\alpha_{k,x} |k\rangle \Big). && (5.27)\end{aligned}$$

Finally, applying S again gives,

$$\sum_x \delta_x \left|x\right\rangle \Big(\left|E_s^k\right\rangle - 2\alpha_{k,x} H_x \left|k\right\rangle \Big) \tag{5.28}$$

$$= \sum_x \delta_x \left|x\right\rangle \Big(\left|E_s^k\right\rangle - 2\alpha_{k,x} \left|E_x^k\right\rangle \Big) \tag{5.29}$$

$$= \sum_x \delta_x (-1)^{x=s} \left|x\right\rangle \Big((-1)^{x=s} \left|E_s^k\right\rangle - (-1)^{x=s} 2\alpha_{k,x} \left|E_x^k\right\rangle \Big) \tag{5.30}$$

$$= \sum_x \delta_x (-1)^{x=s} \left|x\right\rangle \Big(\left|E_s^k\right\rangle - \left|E_s^k\right\rangle + (-1)^{x=s} \left|E_s^k\right\rangle - (-1)^{x=s} 2\alpha_{k,x} \left|E_x^k\right\rangle \Big) \tag{5.31}$$

$$\begin{aligned} &= \Big(\sum_x \delta_x (-1)^{x=s} \left|x\right\rangle \Big) \left|E_s^k\right\rangle \\ &\qquad + \sum_x \delta_x (-1)^{x=s} \left|x\right\rangle \Big(((-1)^{x=s} - 1) \left|E_s^k\right\rangle - (-1)^{x=s} 2\alpha_{k,x} \left|E_x^k\right\rangle \Big). \end{aligned} \tag{5.32}$$

The first term is exactly $(O_s \left|x\right\rangle) \left|E_s^k\right\rangle$, so the distance we are upper bounding is at most the norm of the second term. If we compute the norm-squared of the second term, we get,

$$\begin{aligned} \sum_x |\delta_x|^2 \langle x|x\rangle \Big(& \\ & ((-1)^{x=s} - 1)^2 \langle E_s^k|E_s^k\rangle \\ & + ((-1)^{x=s} 2)^2 |\alpha_{k,x}|^2 \langle E_x^k|E_x^k\rangle \\ & - ((-1)^{x=s} - 1)(-1)^{x=s} 2\alpha_{k,x} \langle E_s^k|E_x^k\rangle \\ & - ((-1)^{x=s} - 1)(-1)^{x=s} 2\alpha_{k,x}^* \langle E_x^k|E_s^k\rangle \\ \Big), & \end{aligned} \tag{5.33}$$

which is equal to,

$$\sum_x |\delta_x|^2\Big(((-1)^{x=s}-1)^2 + 4|\alpha_{k,x}|^2 - 4((-1)^{x=s}-1)(-1)^{x=s}|\alpha_{k,x}|^2\Big) \tag{5.34}$$

$$= \sum_{x\neq s} |\delta_x|^2(0 + 4|\alpha_{x,x}|^2 + 0) + |\delta_s|^2(4 + (4-8)|\alpha_{k,s}|^2) \tag{5.35}$$

$$= \sum_{x\neq s} |\delta_x|^2 4|\alpha_{k,x}|^2 + |\delta_s|^2(4 + (4-8)|\langle E_s^k|E_s^k\rangle|^2) \tag{5.36}$$

$$= \sum_{x\neq s} |\delta_x|^2 4|\alpha_{k,x}|^2 + |\delta_s|^2(4 + (4-8)) \tag{5.37}$$

$$= \sum_{x\neq s} |\delta_x|^2 4|\alpha_{k,x}|^2 + 0. \tag{5.38}$$

So, the distance is upper-bounded by,

$$\sqrt{\sum_{x\neq s} |\delta_x|^2 4|\alpha_{k,x}|^2}. \tag{5.39}$$

If we let $d(x)$ be the Hamming distance between $g(x)$ and $g(s)$, where g is the random oracle used in the definition of PCC, then,

$$\alpha_{k,x} = \langle E_x^k|E_s^k\rangle = \Big(\frac{1}{\sqrt{2}}\Big)^{d(x)}. \tag{5.40}$$

We will show that with high probability, $d(x) > 1.5n$ for all $x \neq s$. For any particular $x \neq s$, we can think of the hamming distance between $g(x)$ and $g(s)$ as the number of successful outcomes of $3n$ experiments that each succeed with probability $1/2$ independently. So, the probability that $d(x) \leq 1.5n$ can be upper-bounded using the Chernoff bound for the binomial distribution that we discussed in Chapter 2. This gives, for all $x \neq s$,

$$\Pr[d(x) \leq 1.5n] \leq \exp\bigg(\frac{-(\lfloor \ell n\rfloor/2 - 1.5n)^2}{\lfloor \ell n\rfloor}\bigg) \tag{5.41}$$

$$\leq \exp\bigg(\frac{-(\ell n/2 - 1.5n - 0.5)^2}{\ell n}\bigg) \tag{5.42}$$

$$= \exp\bigg(\frac{-(\ell n/2 - 1.5n)^2 - (-0.5\ell n/2 + (0.5)(1.5)n + (0.5)^2)}{\ell n}\bigg) \tag{5.43}$$

$$= \exp\bigg(-n(\ell/2 - 1.5)^2/\ell + 0.5/2 - \frac{(0.5)(1.5)}{\ell} - \frac{(0.5)^2}{\ell n}\bigg) \tag{5.44}$$

$$\leq \exp\big(-n(\ell/2 - 1.5)^2/\ell + 0.25\big). \tag{5.45}$$

Therefore, by the union bound, the probability that $d(x) \leq 1.5n$ for *any* $x \neq s$ is at most,

$$2^n \exp\bigl(-n(\ell/2 - 1.5)^2/\ell + 0.25\bigr) = \exp\bigl(\ln(2)n - n(\ell/2 - 1.5)^2/\ell + 0.25\bigr). \tag{5.46}$$

For this this to decay exponentially in n, we need $\ln(2) - (\ell/2 - 1.5)^2/\ell < 0$. Since we've assumed $\ell > 7.59$, this is true (obtained by solving the inequality with WolframAlpha [wolb]). So, finally, we can say that with high probability, the distance relevant to applying the lemma is upper-bounded by,

$$\sqrt{\sum_{x \neq s} |\delta_x|^2 4 |\alpha_{k,x}|^2} = \sqrt{\sum_{x \neq s} |\delta_x|^2 4 (\frac{1}{2})^{d(x)}} \tag{5.47}$$

$$\leq \sqrt{\sum_{x \neq s} |\delta_x|^2 4 (\frac{1}{2})^{1.5n}} \tag{5.48}$$

$$\leq 2\Bigl(\frac{1}{\sqrt{2}}\Bigr)^{1.5n} \sqrt{1} = 2^{1-3n/4}. \tag{5.49}$$

So indeed with high probability the lemma is satisfied for U, $\left|E_s^k\right\rangle$, and $\epsilon = 2^{1-3n/4}$. This completes our proof that for large enough n, there is an $O(2^{n/2} \cdot \mathsf{poly}(n))$-time algorithm that breaks ℓ-PCC for $\ell > 7.59$ with high probability.

□

We've shown that ℓ-PCC for $\ell > 7.59$ can be broken by a quantum attack requiring $2^{n/2}$ time, which is the same amount of time as it would take to Grover-search for an n-bit key if it were unprotected by an OKE scheme, so using ℓ-PCC for $\ell > 7.59$ does not provide any increase in security against general attacks. It's not obvious how to extend our attack to ℓ-PCC for any $\ell < 7.57$ aside from using a better tail bound on the binomial distribution, avoiding the union bound somehow, or changing the "1.5" in "$d(x) > 1.5n$" to something slightly smaller (e.g. 1.01 improves the result to $\ell > 6.15$ [wola]; going less than or equal to 1 makes ϵ too large to cancel out the factor of $2^{n/2}$ from the lemma).

A better bounding strategy would not help show our attack works against 3-PCC, because considering all $x \neq s$, there's a decent chance there's some $x \neq s$ for which $g(x)$ and $g(s)$ will collide on the first n bits. We can expect half of the $2n$ remaining bits to match, making $\langle E_x^k | E_s^k \rangle \gtrapprox (\frac{1}{\sqrt{2}})^{n/2}$, preventing ϵ from becoming small enough to apply

Lemma 5.3.1 if $|\delta_x|^2$ is near 1. One idea is to improve Lemma 5.3.1 so that the function only needs to be approximated for all states $|\phi\rangle$ that have small $|\langle x|\phi\rangle|$ for all $x \neq s$. This would involve a more detailed analysis of the intermediate states of Grover's algorithm.

Superposition-Blocking Functions

Even if ℓ-PCC is broken for smaller ℓ, it may be possible to modify PCC so that it is secure against adversaries who only have access to weak quantum computers. To carry out our attack, the adversary has to compute the non-quantum decryption function $\mathcal{D}$ in superposition. It might be possible to prevent small quantum computers from doing this by changing PCC to apply a "superposition-blocking" function to the key before passing it to $\mathcal{E}$ and $\mathcal{D}$. A superposition-blocking function is any function that behaves like a hash function and is possible to compute on widely-available classical computers but is too expensive to compute on any quantum computers that will exist for as long as security is needed. The computationally-expensive key-derivation functions used for password storage [PJ16] are good candidates to be superposition-blocking functions, especially yescrypt [Pes] with a large read-only-memory input.

Implementing PCC only requires quantum storage that's reliable for the lifetime of the encrypted message and the ability to perform H gates—much less than large-scale general-purpose quantum computing. So, there could be a time in our future when superposition-blocked PCC can be implemented and there aren't any quantum computers powerful enough to break it. We leave the analysis of PCC's security in "weak quantum computer" models like these to future work.

5.4 Related Work

To the best of our knowledge, quantum encryption schemes with properties similar to offline key expanders have not yet been studied. Several schemes for encrypting classical messages into quantum ciphertexts have been studied, including:

- **Quantum ciphers** [DPS04]. Damgård et al. study the situation where Alice wants to send an encrypted message to Bob, but only one-way transmission from Alice to Bob is allowed. This situation is exactly equivalent to encrypting data at rest (Bob is you, in the future, coming back to decrypt your data), so this is the same as the setting OKEs operate in. The difference between our work and theirs is that in theirs, Alice and Bob want perfect secrecy. Their results show that by using their quantum cipher compared to using a classical one-time pad, the adversary has more Shannon uncertainty about the key given a known-plaintext and ciphertext pair, but the min-entropy, i.e. their probability of guessing the key, remains the same. Just like in the classical case, the key needs to be as long as the message for perfect secrecy. They combine their quantum cipher with a non-quantum stream cipher and show that, assuming a conjecture, a polynomial-time classical+quantum algorithm (with heavy limitations on the quantum part) would need more known-plaintext samples to distinguish the encryption from a perfectly secure one. It's not clear how, if at all, this result could help build OKEs. The two quantum ciphers they define might make good candidate OKEs when used with pseudorandom keys instead of truly random ones.

- **Quantum probabilistic encryption based on conjugate coding [YXL12].** Yang et al. define a way to encrypt a classical message into a quantum ciphertext using a short classical key. They show that given an encryption of a random *unknown* plaintext, obtaining information about the key from the ciphertext violates the no-signalling postulate. They conjecture that their scheme protects the message as well. If the study of their scheme can be completed, and it turns out that for equal key lengths their scheme provides security against stronger adversaries than non-quantum probabilistic encryption does, then their scheme could be used to implement an OKE by encrypting a longer random key.

Along the lines of combining quantum cryptography with complexity-theoretic assumptions, Aaronson has studied quantum copy protection [Aar09]. Aaronson recognizes the potential for the uncloneability of quantum information to help accomplish feats that are provably impossible using non-quantum technology, and shows that relative to a quantum oracle, it is possible to copy-protect a function f. This means that it is possible to compute f using a special quantum state, and the state cannot be cloned. As long as f can not be learned from input-output pairs, the owner of the state is the only one who can compute f. Aaronson mentions one cryptographic task where this would be useful: if f is a boolean function for checking whether a password is correct, then a copy-protected f would be an uncloneable way of verifying the password. Unfortunately, it doesn't seem possible to use uncloneable functions to implement OKEs, because in an attack the adversary has access to the output of the OKE and thus the ability to compute any uncloneable functions it may contain.

5.5 Conclusion

This chapter's contributions are:

- The definition of offline key expanders, a candidate construction called Psuedorandom Conjugate Coding (PCC), and a proof that the ℓ-PCC scheme scheme can be broken with high probability using a $O(2^{n/2})$-time quantum attack when $\ell > 7.59$.
- A lemma that establishes sufficient conditions for Grover's algorithm to work even when the function being searched must be computed using a quantum state, and the searcher only has one copy of the state.
- The notion of a superposition-blocking function, which may be applicable elsewhere to defend against weak quantum attacks.

We've left the following questions unresolved:

- For which values of p, t, n, and m do (p, t)-secure n-to-m OKEs exist?
- Can the attack on the ℓ-PCC scheme be improved to work for smaller values of ℓ? Is ℓ-PCC *secure* for any smaller values of ℓ?
- If ℓ-PCC is insecure against $2^{n/2}$-time quantum attacks, then is it secure against weaker attacks, e.g. ones runing in $2^{n/4}$ time? Can we use superposition-blocking functions to make it secure against attackers with even greater limitations?

Offline key expanders are just one way that combining quantum cryptography with non-quantum cryptography could prove useful. A promising area for future research is to explore what other classically-impossible cryptographic goals can be accomplished by combining quantum information with complexity-theoretic assumptions.

Chapter 6

Conclusion

We began in Chapter 2 by introducing some facts from probability theory and then proved Lemma 2.0.1, a fact about binary-valued random variables. We went on to use Lemma 2.0.1 in our study of parallel repetition and concentration bounds in Chapters 3 and 4.

In Chapter 3, we developed a technique for converting parallel repetition theorems into concentration bounds for nonlocal games. We proved new concentration bounds for certain kinds of games using the parallel repetition theorems that are currently available.

In Chapter 4, we proved that the soundness and completeness errors of a quantum interactive proof system can be reduced through simple parallel repetition, eliminating the need to rely on more complicated error-reduction strategies.

Chapters 3 and 4 are are examples of how Lemma 2.0.1 can be used to reduce the problem of winning n threshold games repeated in parallel to the problem of winning one threshold game (which has a slightly higher threshold). We expect this technique to be applicable to other problems that we did not discuss, e.g. reducing the error in multi-prover quantum interactive proof protocols.

References

[Aar09] Scott Aaronson. Quantum copy-protection and quantum money. In *Computational Complexity, 2009. CCC'09. 24th Annual IEEE Conference on*, pages 229–242. IEEE, 2009.

[BB14] Charles H Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theor. Comput. Sci.*, 560(P1):7–11, 2014.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *Annual International Cryptology Conference*, pages 26–45. Springer, 1998.

[BVY15] Mohammad Bavarian, Thomas Vidick, and Henry Yuen. Anchoring games for parallel repetition. *arXiv preprint arXiv:1509.07466*, 2015.

[CS14] André Chailloux and Giannicola Scarpa. Parallel repetition of entangled games with exponential decay via the superposed information cost. In *International Colloquium on Automata, Languages, and Programming*, pages 296–307. Springer, 2014.

[CSUU08] Richard Cleve, William Slofstra, Falk Unger, and Sarvagya Upadhyay. Perfect parallel repetition theorem for quantum XOR proof systems. *Computational Complexity*, 17(2):282–299, 2008.

[CWY15] Kai-Min Chung, Xiaodi Wu, and Henry Yuen. Parallel repetition for entangled k-player games via fast quantum search. In *Proceedings of the 30th Conference on Computational Complexity*, pages 512–536. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015.

[DKA$^+$14] Zakir Durumeric, James Kasten, David Adrian, J Alex Halderman, Michael Bailey, Frank Li, Nicolas Weaver, Johanna Amann, Jethro Beekman, Mathias

Payer, et al. The matter of heartbleed. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 475–488. ACM, 2014.

[DPS04] Ivan Damgård, Thomas Pedersen, and Louis Salvail. On the key-uncertainty of quantum ciphers and the computational security of one-way quantum transmission. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 91–108. Springer, 2004.

[DR13] Joan Daemen and Vincent Rijmen. *The design of Rijndael: AES—the advanced encryption standard*. Springer Science & Business Media, 2013.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[IK10] Russell Impagliazzo and Valentine Kabanets. Constructive proofs of concentration bounds. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 617–631. Springer, 2010.

[JJUW10] Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous. Qip= pspace. *Communications of the ACM*, 53(12):102–109, 2010.

[JUW09] Rahul Jain, Sarvagya Upadhyay, and John Watrous. Two-message quantum interactive proofs are in PSPACE. *arXiv preprint arXiv:0905.1300*, 2009.

[KW00] Alexei Kitaev and John Watrous. Parallelization, amplification, and exponential time simulation of quantum interactive proof systems. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 608–617. ACM, 2000.

[LW15] Cécilia Lancien and Andreas Winter. Parallel repetition and concentration for (sub-) no-signalling games via a flexible constrained de finetti reduction. *arXiv preprint arXiv:1506.07002*, 2015.

[LWW$^+$10] Lars Lydersen, Carlos Wiechers, Christoffer Wittmann, Dominique Elser, Johannes Skaar, and Vadim Makarov. Hacking commercial quantum cryptography systems by tailored bright illumination. *Nature photonics*, 4(10):686–689, 2010.

[MBC$^+$16] Vadim Makarov, Jean-Philippe Bourgoin, Poompong Chaiwongkhot, Mathieu Gagné, Thomas Jennewein, Sarah Kaiser, Raman Kashyap, Matthieu Legré, Carter Minshull, and Shihan Sajeed. Creation of backdoors in quantum communications via laser damage. *Physical Review A*, 94(3):030302, 2016.

[Mos14] Dana Moshkovitz. Parallel repetition from fortification. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 414–423. IEEE, 2014.

[MP12] Abel Molina Prieto. Parallel repetition of prover-verifier quantum interactions. Master's thesis, University of Waterloo, 2012.

[MV04] David A McGrew and John Viega. The security and performance of the galois/counter mode (GCM) of operation. In *International Conference on Cryptology in India*, pages 343–355. Springer, 2004.

[NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, New York, NY, USA, 10th edition, 2011.

[Pes] Alexander Peslyak. yescrypt - scalable KDF and password hashing scheme. http://www.openwall.com/yescrypt/. [Online; accessed July 10, 2018].

[PJ16] Colin Percival and Simon Josefsson. The scrypt password-based key derivation function. Technical report, 2016.

[Rao11] Anup Rao. Parallel repetition in projection games and a concentration bound. *SIAM Journal on Computing*, 40(6):1871–1891, 2011.

[Raz98] Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

[Sha92] Adi Shamir. IP=PSPACE. *Journal of the ACM (JACM)*, 39(4):869–877, 1992.

[Slo17] William Slofstra. The set of quantum correlations is not closed. *arXiv preprint arXiv:1703.08618*, 2017.

[Ung09] Falk Unger. A probabilistic inequality with applications to threshold direct-product theorems. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 221–229. IEEE, 2009.

[Wat18] John Watrous. *The theory of quantum information.* Cambridge University Press, 2018.

[wola] WolframAlpha: $\ln(2) - (x/2 - 1.01)^2/x < 0$. https://www.wolframalpha.com/input/?i=ln(2)+-+(x%2F2+-+1.01)%5E2%2Fx+%3C+0. [Online; accessed July 13, 2018].

[wolb] WolframAlpha: $\ln(2) - (x/2 - 1.5)^2/x < 0$. http://www.wolframalpha.com/input/?i=ln(2)+-+(x%2F2+-+1.5)%5E2%2Fx+%3C+0. [Online; accessed July 11, 2018].

[Yue16] Henry Yuen. A parallel repetition theorem for all entangled games. *arXiv preprint arXiv:1604.04340*, 2016.

[YXL12] Li Yang, Chong Xiang, and Bao Li. Quantum probabilistic encryption scheme based on conjugate coding. *arXiv preprint arXiv:1204.6664*, 2012.