

Quantum-Secure Symmetric-Key Cryptography Based on Hidden Shifts

Gorjan Alagic¹ and Alexander Russell²

¹ Department of Computer Science and Engineering
University of Connecticut
`acr@cse.uconn.edu`

² QMATH, Department of Mathematical Sciences
University of Copenhagen
`galagic@gmail.com`

Abstract. Recent results of Kaplan et al., building on previous work by Kuwakado and Morii, have shown that a wide variety of classically-secure symmetric-key cryptosystems are completely broken when exposed to *quantum CPA attacks*. In such an attack, the quantum adversary has the ability to query the cryptographic functionality in superposition. The vulnerable cryptosystems include the Even-Mansour block cipher, the three-round Feistel network, the Encrypted-CBC-MAC, and many others. In this work, we study simple algebraic adaptations of such schemes that replace $(\mathbb{Z}/2)^n$ addition with operations over alternate finite groups—such as $\mathbb{Z}/2^n$ —and provide evidence that these adaptations are secure against quantum CPA attacks. These adaptations furthermore retain the classical security properties (and basic structural features) enjoyed by the original schemes.

We establish security by treating the (quantum) hardness of the well-studied *Hidden Shift problem* as a basic cryptographic assumption. We observe that this problem has a number of attractive features in this cryptographic context, including random self-reducibility, hardness amplification, and—in many cases of interest—a reduction from the “search version” to the “decisional version.” We then establish, under this assumption, the security of several such hidden-shift adaptations of symmetric-key constructions against quantum CPA attack. We show that a Hidden Shift version of the Even-Mansour block cipher yields a quantum-secure pseudorandom function, and that a Hidden Shift version of the Encrypted CBC-MAC yields a collision-resistant hash function. Finally, we observe that such adaptations frustrate the direct Simon’s algorithm-based attacks in more general circumstances, e.g., Feistel networks and slide attacks.

1 Introduction

The discovery of efficient quantum algorithms for algebraic problems with long-standing roles in cryptography, like factoring and discrete logarithm [27], has led to a systematic re-evaluation of cryptography in the presence of quantum attacks. Such attacks can, for example, recover private keys directly from public keys for many public-key cryptosystems of interest. A 2010 article of Kuwakado and Morii [17] identified a new family of quantum attacks on certain generic constructions of *private-key* cryptosystems. While the attacks rely on similar quantum algorithmic tools (that is, algorithms for the hidden subgroup problem), they qualitatively differ in several other respects. Perhaps most notably, they break reductions which are information-theoretically secure in the classical setting. On the other hand, these attacks rely on a “quantum CPA” setting which permits the quantum adversary to make queries—in superposition—to the relevant cryptosystem.

These quantum CPA attacks have been generalized and expanded to apply to a large family of classical symmetric-key constructions, including Feistel networks, Even-Mansour ciphers, Encrypted-CBC-MACs, tweakable block ciphers, and others [13, 17, 18, 26]. A unifying feature of all these new attacks, however, is an application of Simon’s algorithm for recovering “hidden shifts” in the group $(\mathbb{Z}/2)^n$. Specifically, the attacks exploit an internal application of addition (mod 2) to construct an instance of a hidden shift problem—solving the hidden shift problem then breaks the cryptographic construction. As an illustrative example, consider two (independent) uniformly random functions $P, Q : \{0, 1\}^n \rightarrow \{0, 1\}^n$ and a uniformly random element z of $\{0, 1\}^n$. It is easy to see that no classical algorithm can distinguish the function $(x, y) \mapsto (P(x), Q(y))$ from the function $(x, y) \mapsto (P(x), P(y \oplus z))$ with a polynomial number of queries; this observation directly motivates the classical Even-Mansour block-cipher construction. On the other hand, an efficient quantum algorithm with oracle access to $(x, y) \mapsto (P(x), P(y \oplus z))$ can apply Simon’s algorithm to recover the “hidden shift” z efficiently; this clearly allows the algorithm to distinguish the two cases above.

While these attacks threaten many classical private-key constructions, they depend on an apparent peculiarity of the group $(\mathbb{Z}/2)^n$ —the HIDDEN SHIFT problem over $(\mathbb{Z}/2)^n$ admits an efficient quantum algorithm. In contrast, HIDDEN SHIFT problems in general have resisted over 20 years of persistent attention from the quantum algorithms community. Indeed, aside from Simon’s polynomial-time algorithm for hidden shifts over $(\mathbb{Z}/2)^n$, generalizations to certain groups of constant exponent [9], and Kuperberg’s $2^{O(\sqrt{\log N})}$ algorithm for hidden shifts over \mathbb{Z}/N [16], very little is known. This lack of progress is not due to a lack of motivation: it is well-known that efficient quantum algorithms for HIDDEN SHIFT over \mathbb{Z}/N would yield efficient quantum attacks on important public-key cryptosystems [24, 25], including prime candidates for quantum security and the eventual replacement of RSA in Internet cryptography [3]. Likewise, efficient algorithms for the symmetric group would yield efficient quantum algorithms for GRAPH ISOMORPHISM, a longstanding challenge in the area.

On the other hand, $(\mathbb{Z}/2)^n$ group structure is rather incidental to the security of typical symmetric-key constructions. For example, the classical Even-Mansour

construction defines a block cipher $E_{k_1, k_2}(m)$ by the rule

$$E_{k_1, k_2}(m) = P(m \oplus k_1) \oplus k_2,$$

where P is a public random permutation and the secret key (k_1, k_2) is given by a pair of independent elements drawn uniformly from $(\mathbb{Z}/2)^n$. The security proofs, however, make no particular assumptions about group structure, and apply if “ \oplus ” is replaced with an alternative group operation, e.g., $+$ modulo N or multiplication in \mathbb{F}_{2^n} .

This state of affairs suggests the possibility of ruling out quantum attacks by the simple expedient of adapting the underlying group in the construction. Moreover, the apparently singular features of $(\mathbb{Z}/2)^n$ in the quantum setting suggest that quite mild adaptations may be sufficient. As mentioned above, many classical security proofs are unaffected by this substitution; our primary goal is to add security against quantum adversaries. *Our approach is to reduce well-studied Hidden Shift problems directly to the security of these symmetric-key cryptosystems. Thus, efficient quantum CPA attacks on these systems would resolve long-standing open questions in quantum complexity theory.*

1.1 Contributions

Hidden shift as a cryptographic primitive. We propose the intractability of the HIDDEN SHIFT problem as a fundamental assumption for establishing quantum security of cryptographic schemes. In the general problem, we are given two functions on some finite group G , and a promise that one is a shift of the other; our task is to identify the shift. Our assumptions have the following form:

Assumption 1. *[The \mathcal{G} -Hidden Shift Assumption, informal] Let $\mathcal{G} = \{G_i \mid i \in I\}$ be a family of finite groups indexed by a set $I \subset \{0, 1\}^*$. For all polynomial-time quantum algorithms \mathcal{A} ,*

$$\min_{s \in G_i} \Pr_f \left[\mathcal{A}^{f, f_s}(i) = s \right] \leq \text{negl}(|i|),$$

where $f_s(x) = f(sx)$, the probability is taken over random choice of the function f (and internal randomness and measurements of \mathcal{A}), and the minimum is taken over all shifts $s \in G_i$.

This assumption asserts that there is no quantum algorithm for HIDDEN SHIFT (over \mathcal{G}) in the *worst-case* over s , when function values are chosen randomly. Note that the typical formulation in the quantum computing literature is worst case over s and f ; on the other hand, known algorithmic approaches are invariant under arbitrary relabeling of the value space of f . The “random-valued” case thus seems satisfactory for our cryptographic purposes. (In fact, our results can alternatively depend on the case where f is efficient rather than random.) See [Section 3](#) below for further discussion and precise versions of [Assumption 1](#). In general, formulating such an assumption requires attention to the encoding of the

group. However, we will focus almost entirely on groups with conventional encodings which directly provide for efficient group operations, inversion, generation of random elements, etc. Specifically, we focus on the two following particular variants:

Assumption 2 (The 2^n -Cyclic Hidden Shift Assumption). *This is the Hidden Shift Assumption with the group family $\mathcal{C}_2 = \{\mathbb{Z}/2^n \mid n \geq 0\}$ where the index consists of the number n written in unary.*

Assumption 3 (The Symmetric Hidden Shift Assumption). *This is the Hidden Shift Assumption with the group family $\mathcal{S} = \{S_n \mid n \geq 0\}$ where S_n denotes the symmetric group on n symbols and the index consists of the number n written in unary.*

In both cases the size of the group is exponential in the length of the index.

We remark that the HIDDEN SHIFT problem has polynomial quantum query complexity [6]—thus one cannot hope that HIDDEN-SHIFT-based schemes possess information-theoretic security in the quantum setting (as they do in the classical setting); hence the introduction of HIDDEN SHIFT intractability assumptions.

To explore the hardness of HIDDEN SHIFT problems against quantum polynomial-time (QPT) algorithms, we describe several reductions. First, we prove that HIDDEN SHIFT is equivalent to a randomized version of the problem where the shift s is random (RANDOM HIDDEN SHIFT), and provide an amplification theorem which is useful in establishing security of schemes based on Assumption 1. The theorem requires the existence of quantum-secure pseudorandom functions (which can be constructed from quantum-secure one-way functions [29].)

Proposition (Amplification, informal). *Assume there exists a QPT which solves RANDOM HIDDEN SHIFT for an inverse-polynomial fraction of inputs. Then there exists a QPT for solving both HIDDEN SHIFT and RANDOM HIDDEN SHIFT for all but a negligible fraction of inputs.*

We then show that, for many group families, HIDDEN SHIFT over the relevant groups is equivalent to a *decisional* version of the problem. In the decisional version, we are guaranteed that the two functions are either (i.) both random and independent, or (ii.) one is random and the other is a shift; the goal is to decide which is the case.

Theorem (Search and decision are equivalent, informal). *Let \mathcal{G} be the group family \mathcal{C}_2 or the group family \mathcal{S} (or a group family with an efficient subgroup series). Then there exists a QPT for RANDOM HIDDEN SHIFT (with at most inverse-poly error) over \mathcal{G} if and only if there exists a QPT for DECISIONAL RANDOM HIDDEN SHIFT (with at most inverse-poly error) over \mathcal{G} .*

Finally, we provide some evidence that the HIDDEN SHIFT over the family \mathcal{C}_2 is as hard as HIDDEN SHIFT over general cyclic groups. Specifically, we show that efficient algorithms for an approximate version of HIDDEN SHIFT over \mathcal{C}_2

give rise to efficient algorithms for the same problem over \mathcal{C} , the family of all cyclic groups.

We also briefly discuss the connections between HIDDEN SHIFT, the assumptions above, and assumptions underlying certain candidates for quantum-secure public-key cryptography [5, 24]. For completeness, we also briefly recall known connections to the HIDDEN SUBGROUP PROBLEM. Both the HIDDEN SHIFT and HIDDEN SUBGROUP PROBLEM families have received significant attention from the quantum algorithms community, and are believed to be quantumly hard with the exception of particular families of groups [5, 12, 20, 21, 24].

Quantum-secure symmetric-key cryptographic schemes. With the above results in hand, we describe a generic method for using Assumption 1 to “adapt” classically-secure schemes in order to remove any vulnerabilities to quantum CPA attacks. The adaptation is simple: replace the underlying $(\mathbb{Z}/2)^n$ structure of the scheme with that of either \mathcal{C}_2 or \mathcal{S} . This amounts to replacing bitwise XOR with a new group operation. In the case of \mathcal{C}_2 , the adaptation is particularly simple and efficient.

While our basic approach presumably applies in broad generality, we focus on three emblematic examples: the Even-Mansour construction—both as a PRF and as a block cipher—and the CBC-MAC construction. We focus throughout on the group families \mathcal{C}_2 and \mathcal{S} , though we also discuss some potential advantages of other choices (see Section 3.2.) Finally, we discuss related quantum attacks on cryptographic constructions, including the 3-round Feistel cipher and so-called quantum slide attacks [13]. We remark that the Feistel cipher over groups other than $(\mathbb{Z}/2)^n$ has been considered before, in a purely classical setting [22].

Hidden Shift Even-Mansour. Following the prescription above, we define group variants of the Even-Mansour cipher. We give a reduction from the worst-case HIDDEN SHIFT problem to the natural *distinguishability* problem (i.e., distinguishing an Even-Mansour cipher from a random permutation). Thus, under the Hidden Shift Assumption, the Even-Mansour construction is a qPRF. In particular, key-recovery is computationally infeasible, even for a quantum adversary. We also provide (weaker) reductions between HIDDEN SHIFT and the problem of breaking Even-Mansour in the more challenging case where the adversary gets access to both the public permutation and its inverse (and likewise for the encryption map). We remark that, in any case, these adaptations frustrate the “Simon algorithm key recovery attack” [18, 13], as this would now require a subroutine for HIDDEN SHIFT in the relevant group family. Moreover, one can also apply standard results (see, e.g., [11]) to show that, over some groups, all bits of the key are as hard as the entire key (and hence, by our reductions, as hard as HIDDEN SHIFT).

Hidden Shift CBC-MAC. Following our generic method for transforming schemes, we define group variants of the Encrypted-CBC-MAC. We establish that this primitive is collision-free against quantum adversaries. Specifically, we show that

any efficient quantum algorithm which discovers collisions in the Hidden-Shift Encrypted-CBC-MAC with non-negligible probability would yield an efficient worst-case quantum algorithm for HIDDEN SHIFT over the relevant group family. As with Even-Mansour, this adaptation immediately frustrates the Simon’s algorithm collision-finding attacks [13, 26].

Feistel ciphers, slide attacks. We also define group variants of the well-known Feistel cipher for constructing pseudorandom permutations from pseudorandom functions. Our group variants frustrate Simon-style attacks [17]; a subroutine for the more general HIDDEN SHIFT problem is now required. Finally, we also address the exponential quantum speedup of certain classical slide attacks, as described in [13]. We show how one can once again use HIDDEN SHIFT to secure schemes vulnerable to these “quantum slide attacks.”

2 Preliminaries

Notation; remarks on finite groups. For a finite group G and an element $s \in G$, let $L_s : G \rightarrow G$ denote the permutation given by left multiplication by s , so $L_s : x \mapsto s \cdot x$. We discuss a number of constructions in the paper requiring computation in finite groups and assume, throughout, that elements of the group in question have an encoding that efficiently permits such natural operations as product, inverse, selection of uniformly random group elements, etc. As our discussion focuses either on specific groups—such as $(\mathbb{Z}/2)^n$ or \mathbb{Z}/N —where such encoding issues are straightforward or, alternatively, *generic* groups on which we make no specific structural claims, we routinely ignore these issues of encoding. We revisit this issue in [Section 3.2](#).

Classical and quantum algorithms. Throughout we use the abbreviation PPT for “probabilistic polynomial time,” referring to an efficient classical algorithm, and QPT for “quantum polynomial time,” referring to an efficient quantum algorithm. Our convention is to denote algorithms of either kind with script letters, e.g., \mathcal{A} will typically denote an algorithm which models an adversary. If $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ is a function family, the notation \mathcal{A}^f stands for an algorithm (either classical or quantum) with oracle access to the function f . A classical oracle is simply the black-box gate $x \mapsto f(x)$; a quantum oracle is the unitary black-box gate $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$. Unless stated otherwise, oracle QPTs are assumed to have quantum oracle access.

Quantum-secure pseudorandomness. We now set down a way of quantifying the ability of a QPT adversary to distinguish an efficiently sampled function from a uniformly random one. Fix a function family $\mathcal{F} \subset \{h : \{0, 1\}^m \rightarrow \{0, 1\}^\ell\}$, and a function $f : \{0, 1\}^n \times \{0, 1\}^m \rightarrow \{0, 1\}^\ell$, and define $f_k := f(k, \cdot)$. We say that f is an *indexed subfamily* of \mathcal{F} if $f_k \in \mathcal{F}$ for every $k \in \{0, 1\}^n$. We will generally assume that m and ℓ are polynomial functions of n ; we take n to be the complexity (or security) parameter.

Definition 1. Let \mathcal{F} be a function family, f an indexed subfamily, and \mathcal{D} an oracle QPT. The distinguishing advantage of \mathcal{D} is the quantity

$$\mathbf{Adv}_{\mathcal{F},f}^{\mathcal{D}} := \left| \Pr_{k \in_R \{0,1\}^n} [\mathcal{D}^{f_k}(1^n) = 1] - \Pr_{g \in_R \mathcal{F}} [\mathcal{D}^g(1^n) = 1] \right|.$$

Next, we define efficient indexed function families which are pseudorandom against QPT adversaries. We emphasize that these function families are computed by deterministic classical algorithms.

Definition 2. Let \mathcal{F}_n be the family of all functions from $m(n)$ bits to $\ell(n)$ bits, and f a efficiently computable, indexed subfamily of $\bigcup_n \mathcal{F}_n$ (so that $f_k \in \mathcal{F}_n$ for $|k| = n$). We say that f is a quantum-secure pseudorandom function (qPRF) if $\mathbf{Adv}_{\mathcal{F}_n,f}^{\mathcal{D}} \leq \text{negl}(n)$ for all QPTs \mathcal{D} .

It is known how to construct qPRFs from standard assumptions (i.e., existence of quantum-secure one-way functions) [29]. We will frequently use the assumption that qPRFs exist when we discuss the relationship between various hardness assumptions.

The pseudorandom function property is not enough in certain applications, e.g., in building block ciphers. It is often useful to add the property that each function in the family is a permutation, which can be inverted efficiently (provided the index is known).

Definition 3. Let \mathcal{P} be the family of all permutations, and f an efficiently computable, indexed subfamily of \mathcal{P} . We say that f is a quantum-secure pseudorandom permutation (qPRP) if (i.) f is a qPRF, (ii.) each f_k is a permutation, and (iii.) there is an efficient algorithm which, given k , computes the inverse f_k^{-1} of f_k .

To our knowledge, there are currently no known constructions for quantum-secure pseudorandom permutations. Two constructions which are known to work classically, Even-Mansour and the 3-round Feistel, are both broken by the Simon's algorithm attack mentioned above. As we discuss in detail later, we conjecture that the adaptations of these constructions to other group families are qPRPs.

Collision-freeness. We will also need a (standard) definition of collision-resistance against efficient quantum adversaries with oracle access.

Definition 4. Let $f : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ be an efficiently-computable function family defined for all (k,x) for which $|x| = m(|k|)$ (for a polynomial m). We say that f is collision-resistant if for all QPTs \mathcal{A} ,

$$\Pr_{k \in_R \{0,1\}^n} [\mathcal{A}^{f_k}(1^n) = (x,y) \wedge f_k(x) = f_k(y)] \leq \text{negl}(n).$$

3 Hidden Shift as a cryptographic primitive

We begin by discussing a few versions of the basic oracle promise problem related to finding hidden shifts of functions on groups. In the problems below, the relevant functions are given to the algorithm via black-box oracle access and we are interested in the setting where the complexity of the algorithm (both number of queries and running time) scales in $\text{poly}(\log |G|)$.

3.1 Hidden Shift problems

Basic definitions. We begin with the HIDDEN SHIFT problem. As traditionally formulated in the quantum computing literature, the problem is the following:

Problem 1 (The traditional Hidden Shift problem). Let G be a group and V a set. Given oracle access to an injective function $f : G \rightarrow V$ and an unknown shift $g = f \circ L_s$ of f , find s .

It is convenient for us to parameterize this definition in terms of a specific group family and fix the range of the oracles f and g . This yields our basic asymptotic definition for the problem.

Problem 2 (HIDDEN SHIFT (HS)). Let $\mathcal{G} = \{G_i \mid i \in I\}$ be a family of groups for an index set $I \subset \{0, 1\}^*$ and let $\ell : \mathbb{N} \rightarrow \mathbb{N}$ be a polynomial. Then the HIDDEN SHIFT problem over \mathcal{G} (with length parameter ℓ) is the following: given an index i and oracle access to a pair of functions $f, g : G_i \rightarrow \{0, 1\}^{\ell(|i|)}$ where $g(x) = f(sx)$, determine $s \in G_i$. We assume, throughout, that $2^{\ell(|i|)} \gg |G_i|$.

This generic formulation is more precise, but technically still awkward for cryptographic purposes as it permits oracle access to completely arbitrary functions f . This interferes with integrating the problem with other cryptographic primitives (e.g., qPRFs) because the functions could, in principle, leak information about the other primitives. We discuss this in more detail below.

To avoid this technical obstacle, we focus on the performance of HIDDEN SHIFT algorithms over specific classes of functions f . Specifically, we either assume f is random or, alternatively, generated by an efficient algorithm. Of course, when a HIDDEN SHIFT algorithm is applied to solve problems in a computational setting, the actual functions f, g are given by efficient computations. We remark that typical algorithmic practice in this area ignores the actual function values, merely relying on the structure of the level sets of the function

$$\Phi(x, b) = \begin{cases} f(x) & \text{if } b = 0, \\ g(x) & \text{if } b = 1. \end{cases}$$

In particular, such structural conditions of f appear to be irrelevant to the success of current quantum algorithmic techniques for the problem. This motivates the following notion of “success” for an algorithm.

Definition 5 (Completeness). Let \mathcal{A} be an algorithm for the HIDDEN SHIFT problem on \mathcal{G} with length parameter ℓ . Then define

$$1 - \epsilon_R(i) = \min_{s \in G_i} \Pr[\mathcal{A}^{f, f_s}(i) = s],$$

where $f : G_i \rightarrow \{0, 1\}^{\ell(|i|)}$ is drawn uniformly at random. We say that \mathcal{A} has completeness $1 - \epsilon_R(i)$ for random functions. Note that this notion of complexity is worst-case in s , the shift.

Similarly, let f be a polynomial-time computable injective function defined on all pairs (i, x) where $x \in G_i$ so that $f(i, x) \in \{0, 1\}^{\ell(n)}$. Then we define

$$1 - \epsilon_f(i) = \min_{s \in G_i} \Pr[\mathcal{A}^{f, f_s}(i) = s].$$

This gives a notion of completeness relative to f .

Note that this definition does not specify how the algorithm should behave on instances that are not hidden shifts. For simplicity, we assume that the algorithm returns a value for s in any case, with no particular guarantee on s in the case when the functions are not shifts of each other.

As mentioned in the introduction, a precise description of the problem in general would require stipulations about the encodings of group elements and efficiency of group operations with respect to this encoding (e.g., insisting that the groups G_i are efficient black-box groups with unique encodings). As we restrict our attention to specific group families (e.g., \mathcal{C}_2 and \mathcal{S}) with canonical, concrete encodings, we don't set down a specific general definition.

Our basic hardness assumption is the following:

Assumption 4 (The \mathcal{G} -Hidden Shift Assumption; randomized). Let $\mathcal{G} = \{G_i \mid i \in I\}$ be a family of finite groups indexed by a set $I \subset \{0, 1\}^*$ and $\ell : \mathbb{N} \rightarrow \mathbb{N}$ be a length parameter. Then for all efficient algorithms \mathcal{A} , $1 - \epsilon_R(i) = \text{negl}(|i|)$.

For completeness, we also record a version of the assumption for efficient f ; we will see shortly that this is a stronger assumption. In practice, our cryptographic constructions will rely only on the randomized version.

Assumption 5 (The \mathcal{G} -Hidden Shift Assumption; efficient). Let $\mathcal{G} = \{G_i \mid i \in I\}$ be a family of finite groups indexed by a set $I \subset \{0, 1\}^*$ and $\ell : \mathbb{N} \rightarrow \mathbb{N}$ be a length parameter. Then for all efficient algorithms \mathcal{A} there exists an efficient f (satisfying the criteria of [Definition 5](#) above), so that $1 - \epsilon_f(i) = \text{negl}(|i|)$.

In preparation for establishing results on security amplification, we define two additional variants of the HIDDEN SHIFT problem: a variant where both the function and the shift are randomized, and a decisional variant. Our general approach for constructing security proofs will be to reduce one of these variants to the problem of breaking the relevant cryptographic scheme. As we will later show, an efficient solution to either variant implies an efficient solution to both, which in turn results in a violation of [Assumption 4](#) above.

Problem 3 (RANDOM HIDDEN SHIFT (RHS)). Let $\mathcal{G} = \{G_i \mid i \in I\}$ be a group family and $\ell(i)$ a length parameter. Then the RANDOM HIDDEN SHIFT problem over \mathcal{G} is the HIDDEN SHIFT problem where the input function $f : G_i \rightarrow \{0, 1\}^{\ell(i)}$ is drawn uniformly among all functions and the shift s is drawn (independently and uniformly) from G_i .

We define the completeness $1 - \epsilon(i)$ for a RANDOM HIDDEN SHIFT algorithm \mathcal{A} analogously to [Definition 5](#). Observe that a small error is unavoidable for any algorithm, as there exist pairs of functions for which s is not uniquely defined. We will also need a decision version of the above, defined as follows.

Problem 4 (DECISIONAL RANDOM HIDDEN SHIFT (DRHS)). Let \mathcal{G} be a group family and $\ell(i)$ a length parameter. The DECISIONAL RANDOM HIDDEN SHIFT problem is the following: Given i and oracle access to two functions $f, g : G \rightarrow \{0, 1\}^{\ell(i)}$ with the promise that either (i.) both f and g are drawn independently at random, or (ii.) f is random and $g = f \circ L_s$ for some $s \in G$, decide which is the case.

We say that an algorithm for DRHS has completeness $1 - \epsilon(i)$ and soundness $\delta(i)$ if the algorithm errs with probability no more than $\epsilon(i)$ in the case that the functions are shifts and errs with probability no more than $\delta(i)$ in the case that the functions are drawn independently.

Next, we briefly recall the definition of the (closely-related) HIDDEN SUBGROUP PROBLEM. For us, this problem is relevant primarily because of its historical significance (and relationship to HIDDEN SHIFT); we will not use it directly in any security reductions.

Problem 5 (HIDDEN SUBGROUP PROBLEM (HSP)). Let G be a group and S a set. Given a function $f : G \rightarrow S$, and a promise that there exists $H \leq G$ such that f is constant and distinct on the right cosets of H , output a complete set of generators for H .

Some further details, including explicit reductions between HS and HSP, are given in [Appendix A](#).

Of interest are both classical and quantum algorithms for solving the various versions of HS and HSP. The relevant metrics for such algorithms are the query complexity (i.e., the number of times that the functions are queried, classically or quantumly) as well as their time and space complexity. An algorithm is said to be efficient if all three are polynomial in $\log |G|$. This is the “right quantity” for several reasons; note, for example, that one can always give a generating set for G of size $O(\log |G|)$.

Hardness results. Next, we establish several reductions between these problems. Roughly, these results show that the average-case and decisional versions of the problem are as hard as the worst-case version.

Self-reducibility and amplification. First, we show that, if qPRFs exist, then (i.) both HS and RHS are random self-reducible, and (ii.) an efficient solution to RHS implies an efficient solution to HS.

Proposition 1. *Fix a group family $\mathcal{G} = \{G_i\}$ and a length parameter $\ell(i)$. Assume there exists a QPT \mathcal{A} which solves RANDOM HIDDEN SHIFT over \mathcal{G} (with parameter $\ell(i)$) with inverse-polynomial completeness. Then, if qPRFs exist there exists a QPT \mathcal{A}' which satisfies all of the following:*

1. \mathcal{A}' solves HIDDEN SHIFT with random f with completeness $1 - \text{negl}(n)$;
2. \mathcal{A}' solves HIDDEN SHIFT for any efficient f with completeness $1 - \text{negl}(n)$;
3. \mathcal{A}' solves RANDOM HIDDEN SHIFT with completeness $1 - \text{negl}(n)$.

Proof. Let F be a qPRF. We are given oracles f, g and a promise that $g = f \circ L_s$. The algorithm \mathcal{A}' will repeatedly execute the following subroutine. First, we select a key k for F and a uniformly random $t \in G_i$. We then execute \mathcal{A} with oracles

$$f' := F_k \circ f \quad \text{and} \quad g' := F_k \circ g \circ L_t.$$

It's easy to see that $g' = f' \circ L_{st}$. If \mathcal{A} outputs a group element r , we check if $g'(x) = f'(rx)$ at a polynomial number of different values x . If the check succeeds, we output rt^{-1} and terminate \mathcal{A}' . If the check fails (or if \mathcal{A} outputs garbage), we say the subroutine fails. We repeat the subroutine m times, each time with a fresh k and t .

We remark that \mathcal{A} , when used as a subroutine as above, cannot distinguish between the case where F_k is sampled from a qPRF, and the case where it is replaced by a truly random function. This is not completely trivial, due to the presence of oracles. Assume for a contradiction that \mathcal{A} can distinguish between these two cases. If the instance f, g is from a HIDDEN SHIFT problem for efficient functions (case 2 above), then we can replace the oracles with explicit circuits for computing f and g , thus turning \mathcal{A} into a complete (non-oracle) distinguisher between F_k and a random function. In the case where f, g are random functions (cases 1 and 3), note that \mathcal{A} is receiving evaluations of F_k on random inputs. Clearly this is weaker than the usual setting of distinguishability (where \mathcal{A} can in fact choose inputs to F_k), and the claim still holds.

Now, let us prove that \mathcal{A}' satisfies the claims 1-3. If f is random (cases 1 and 3) or injective (case 2), then f' is indistinguishable from random. In either case, g' is a shift of f' by st , and st is distributed uniformly. It follows that, with inverse-polynomial probability ϵ (over the choice of k and t), the instance (f', g') is indistinguishable from an instance (φ, φ_{st}) on which the subroutine succeeds with inverse-polynomial probability δ . After m repetitions of the subroutine, \mathcal{A}' will correctly compute the shift $r = st$ with probability at least $(1 - \epsilon\delta)^m \approx e^{-\epsilon\delta m}$, as desired. \square

Decision versus search. Next, we consider the relationship between searching for shifts (given the promise that one exists,) and deciding if a shift exists or not. Roughly speaking, we establish that the two problems are equivalent for

most group families of interest. We begin with a straightforward reduction from DRHS to RHS.

Proposition 2. *If there exists a QPT for RANDOM HIDDEN SHIFT on \mathcal{G} with completeness $1 - \epsilon(i)$, then there exists a QPT for DECISIONAL RANDOM HIDDEN SHIFT on \mathcal{G} with completeness $1 - \epsilon(i)$, and negligible soundness error.*

Proof. Let $\ell(i)$ be the relevant length parameter. Consider an RHS algorithm for G with completeness $1 - \epsilon$ and the following adaptation to DRHS.

- Run the RHS algorithm.
- When the algorithm returns a purported shift s , check s for veracity with a polynomial number of (classical) oracle queries to f and g (ensuring that $g(x_i) = f(sx_i)$ for $k(n)$ distinct samples $x_1, \dots, x_{k(n)}$).

Observe that if f and g are indeed hidden shifts, this procedure will determine that with probability $1 - \epsilon$. When f and g are unrelated random functions, the “testing” portion of the algorithm will erroneously succeed with probability no more than $|G| \cdot 2^{-k \cdot \ell(i)}$. Thus, under the assumption that $|G| \geq k(n)$, the resulting DRHS algorithm has completeness $1 - \epsilon$ and soundness $|G| \cdot 2^{-k \cdot \ell(i)}$. For any nontrivial length function ℓ , this soundness can be driven exponentially close to zero by choosing $k = \log |G| + k'$. \square

On the other hand, we are only aware of reductions from RHS to DRHS under various structural assumptions on the group G . For example, an algorithmic approach of Fenner and Zhang [8] can be adapted to provide a reduction for the case when G has a “dense” tower of subgroups (and both S_n and $\mathbb{Z}/2^n$ have such towers).

Proposition 3. *Let \mathcal{G} be either the group family $\{\mathbb{Z}/2^n\}$, or the group family $\{S_n\}$. If there exists a QPT for DECISIONAL RANDOM HIDDEN SHIFT on \mathcal{G} with at most inverse-polynomial completeness and soundness errors, then there exists a QPT for RANDOM HIDDEN SHIFT on \mathcal{G} with negligible completeness error.*

Proof. The proof adapts techniques of [8] to our probabilistic setting, and relies on the fact that these group families have an efficient subgroup tower. Specifically, each G_i possesses a subgroup series $\{1\} = G^{(0)} < G^{(1)} < G^{(2)} < \dots < G^{(s)} = G_i$ for which (i.) uniformly random sampling and membership in $G^{(t)}$ can be performed efficiently for all t , and (ii.) for all t , there is an efficient algorithm for producing a left transversal of $G^{(t-1)}$ in $G^{(t)}$. For $\mathbb{Z}/2^n$, the subgroup series is $\{1\} < \mathbb{Z}/2 < \mathbb{Z}/2^2 < \mathbb{Z}/2^3 < \dots$. For S_n (i.e., the group of permutations of n letters), the subgroup series is $\{1\} < S_1 < S_2 < S_3 < \dots$, where each step of the series adds a new letter. We remark that such series can be efficiently computed for general permutation groups using a *strong generating set*, which can be efficiently computed from a presentation of the group in terms of generating permutations [10].

We recursively define a RHS algorithm by considering the case of a group G with a subgroup H of polynomial index with a known left transversal $A =$

$\{a_1, \dots, a_k\}$ (so that G is the disjoint union of the $a_i H$). Assume that the DRHS algorithm for H has completeness δ_H and soundness $1 - \epsilon_H$. In this case, the algorithm (for G) may proceed as follows:

1. For each $\alpha \in A$, run the DRHS algorithm on the two functions f and $\check{g} : x \mapsto g(\alpha x)$ restricted to the subgroup H .
2. If exactly one of these recursive calls reports that the function f and $x \mapsto g(\alpha x)$ are hidden shifts, recursively apply the RHS algorithm to recover the hidden shift s' (so that $f(x) = g(\alpha s' x)$ for $x \in H$). Return the shift $s = \alpha s'$.
3. Otherwise assert that the functions are unrelated random functions.

In the case that f and g are independent random functions, the algorithm above errs with probability no more than $[G : H]\delta_H$.

Consider instead the case that $f : G \rightarrow S$ is a random function and $g(x) = f(sx)$ for an element $s \in G$. Observe that if $s^{-1} \in \alpha_i H$, so that $s^{-1} = \alpha_i h_s$ for an element $h_s \in H$, we have $g(\alpha_i h_s x) = f(x)$. It follows that f and $\check{g} : x \mapsto g(\alpha_i x)$ are shifts of each other; in particular, this is true when restricted to the subgroup H . Moreover, the hidden shift s can be determined directly from the hidden shift between f and \check{g} . Note that, as above, the probability that any of the recursive calls to DRHS are answered incorrectly is no more than $[G : H]\delta_H + \epsilon_H$.

It remains to analyze the completeness of the resulting recursive RHS algorithm: in the case of the subgroup chain above, let γ_t denote the completeness of the resulting RHS algorithm on $G^{(t+1)}$ and note that

$$\gamma_{t+1} \leq [G^{(t+1)} : G^{(t)}]\delta_{G^{(t)}} + \epsilon_{G^{(t)}} + \gamma_t$$

and thus that the resulting error on G is no more than

$$\sum_t [G^{(t+1)} : G^{(t)}]\delta_{G^{(t)}} + \sum_t \epsilon_{G^{(t)}}. \quad (3.1)$$

As mentioned above, both the group families $\{\mathbb{Z}/2^n \mid n \geq 0\}$ and $\{S_n \mid n \geq 0\}$ satisfy this subgroup chain property. \square

Remark. Note that the groups \mathbb{Z}/N for general N are not treated by the results above; indeed, when N is prime, there is no nontrivial tower of subgroups. (Such groups do have other relevant self-reducibility and amplification properties [11].) We remark, however, that a generalization of the HIDDEN SHIFT problem which permits *approximate equality* results in a tight relationship between HIDDEN SHIFT problems for different cyclic groups. In particular, consider the δ -APPROXIMATE HIDDEN SHIFT problem given by two functions $f, g : G \rightarrow S$ with the promise that there exists an element $s \in G$ so that $\Pr_x[g(x) = f(sx)] \geq 1 - \delta$ (where x is chosen uniformly in G); the problem is to identify an element $s' \in G$ with this property. Note that s' may not be unique in this case.

In particular, consider an instance $f, g : \mathbb{Z}/n \rightarrow V$ of a HIDDEN SHIFT problem on a cyclic group \mathbb{Z}/n . We wish to “lift” this instance to a group \mathbb{Z}/m for $m \gg n$ in such a way that a solution to the \mathbb{Z}/m instance yields a solution to the \mathbb{Z}/n

instance. For a function $\phi : \mathbb{Z}/n \rightarrow V$, define the function $\hat{\phi} : \mathbb{Z}/m \rightarrow V$ by the rule $\hat{\phi}(x) = \phi(x \bmod n)$. Note, then, that $\Pr_x[\hat{f}(x) = \hat{g}(\hat{s} + x)] \geq 1 - n/m$ for the shift $\hat{s} = s$; moreover, recovering *any* shift for the \mathbb{Z}/m problem which achieves equality with probability near $1 - n/m$ yields a solution to the \mathbb{Z}/n problem (by taking the answer modulo n , perhaps after correcting for the $m \bmod n$ overhang at the end of the \mathbb{Z}/m oracle). Note that this function is not injective.

3.2 Selecting hard groups

Efficiently solvable cases. For some choices of underlying group G , some of the above problems admit polynomial-time algorithms. A notable case is the HIDDEN SUBGROUP PROBLEM on $G = \mathbb{Z}$, which can be solved efficiently by Shor’s algorithm [27]. The HSP with arbitrary abelian G also admits a polynomial-time algorithm [14]. The earliest and simplest example was Simon’s algorithm [28], which efficiently solves the HSP in the case $G = (\mathbb{Z}/2)^n$ and $H = \{1, s\}$ for unknown s , with only $O(n)$ queries to the oracle. Due to the fact that $(\mathbb{Z}/2)^n \rtimes \mathbb{Z}/2 \cong (\mathbb{Z}/2)^{n+1}$, Simon’s algorithm also solves the HIDDEN SHIFT problem on $(\mathbb{Z}/2)^n$. Additionally, Friedl et al. [9] have given efficient (or quasi-polynomial) algorithms for hidden shifts over solvable groups of constant exponent; for example, their techniques yield efficient algorithms for the groups $(\mathbb{Z}/p)^n$ (for constant p) and $(S_4)^n$.

Cyclic groups. In contrast with the HIDDEN SUBGROUP PROBLEM, the general abelian HIDDEN SHIFT is believed to be hard. The only nontrivial algorithm known is due to Kuperberg, who gave a subexponential-time algorithm for the HSP on dihedral groups [16]. He also gave a generalization to the abelian HIDDEN SHIFT problem, as follows.

Theorem 1. (Theorem 7.1 in [16]) *The abelian HIDDEN SHIFT problem has a quantum algorithm with time and query complexity $2^{O((\log |G|)^{1/2})}$, uniformly for all finitely-generated abelian groups.*

Regev and Kuperberg later improved the above algorithm (so it uses polynomial quantum space, and gains various knobs for tuning complexity parameters), but the time and query complexity remains the same [15, 23].

There is also evidence connecting HSP on the dihedral group D_N (and hence also HS on \mathbb{Z}/N) to other hard problems. Regev showed that, if there exists an efficient quantum algorithm for the dihedral HSP which uses coset sampling (the only nontrivial technique known), then there’s an efficient quantum algorithm for poly(n)-unique-SVP [24]. This problem, in turn, is the basis of several lattice-based cryptosystems. However, due to the costs incurred in the reduction, Kuperberg’s algorithm only yields exponential-time attacks. An efficient solution to HS on \mathbb{Z}/N could also be used to break a certain isogeny-based cryptosystem [4].

We will focus particularly on the case $\mathbb{Z}/2^n$. This is the simplest group for which all of our constructions and results apply. Moreover, basic computational tasks (encoding/decoding group elements as bitstrings, sampling uniformly

random group elements, performing basic group operations, etc.) all have straightforward and extremely efficient implementations over $\mathbb{Z}/2^n$. The existence of a quantum attack with complexity $2^{O(\sqrt{n})}$ in this case will only become practically relevant in the very long term, when the costs of quantum and classical computations become somewhat comparable. If such attacks are truly a concern, then there are other natural group choices, as we discuss below.

Permutation groups. In the search for quantum algorithms for HSP and HS, arguably the most-studied group family is the family of symmetric groups S_n . It is well-known that an algorithm for HSP over $S_n \wr (\mathbb{Z}/2)$ would yield a polynomial-time quantum algorithm for Graph Isomorphism. As discussed in [Appendix A](#), this is precisely the case of HSP relevant to the HIDDEN SHIFT problem over S_n .

For these groups, the efforts of the quantum algorithms community have so far amounted only to negative results. First, it was shown that the standard Shor-type approach of computing with individual “coset states” cannot succeed [21]. In fact, entangled measurements over $\Omega(n \log n)$ coset states are needed [12], matching the information-theoretic upper bound [6]. Finally, the only nontrivial technique for performing entangled measurements over multiple registers, the so-called Kuperberg sieve, is doomed to fail as well [20].

While encoding, decoding, and computing over the symmetric groups is more complicated and less efficient than the cyclic case, it is a well-understood subject (see, e.g., [10]). When discussing these groups below, we will assume (without explicit mention) an efficient solution to these problems.

Matrix groups. Another relevant family of groups are the matrix groups $\text{GL}_2(\mathbb{F}_q)$ and $\text{SL}_2(\mathbb{F}_q)$ over finite fields. These nonabelian groups exhibit many structural features which are similar to the symmetric groups, such as high-dimensional irreducible representations. Many of the negative results concerning the symmetric groups also carry over to matrix groups [12, 20].

Efficient encoding, decoding, and computation over finite fields \mathbb{F}_q is standard. Given these ingredients, extending to matrix groups is not complicated. In the case of $\text{GL}_2(\mathbb{F}_q)$, we can encode an arbitrary pair (not both zero) $(a, c) \in \mathbb{F}_q^2$ in the first column, and any pair (b, d) which is not a multiple of (a, c) in the second column. For $\text{SL}_2(\mathbb{F}_q)$, we simply have the additional constraint that d is fixed to $a^{-1}(1 + bc)$ by the choices of a, b, c .

Product groups. Arguably the simplest group family for which the negative results of [12] apply, are certain n -fold product groups. These are groups of the form G^n where G is a fixed, constant-size group (e.g., S_5 .) This opens up the possibility of simply replacing the XOR operation (i.e., $\mathbb{Z}/2$ addition) with composition in some other constant-size group (e.g., S_5), and retaining the same n -fold product structure.

Some care is needed, however, because there do exist nontrivial algorithms in this case. When the base group G is solvable, then there are efficient algorithms

for both HSP and HIDDEN SHIFT (see Theorem 4.17 in [9]). It is important to note that this efficient algorithm applies even to some groups (e.g., $(S_4)^n$) for which the negative results of [12] also apply. Nevertheless, solvability seems crucial for [9], and choosing $G = S_5$ for the base group gives a family for which no nontrivial HIDDEN SHIFT algorithms are known. We remark that there is however a $2^{O(\sqrt{n \log n})}$ -time algorithm for order-2 HIDDEN SUBGROUP PROBLEMS on G^n based on Kuperberg’s sieve [1]; this suggests the possibility of subexponential (i.e., $2^{O(n^\delta)}$ for $\delta < 1$) algorithms for HIDDEN SHIFT over these groups.

4 Hidden Shift Even-Mansour ciphers

We now address the question of repairing classical symmetric-key schemes which are vulnerable to Simon’s algorithm. We begin with the simplest construction, the so-called Even-Mansour cipher [7].

4.1 Generalizing the Even-Mansour scheme

The standard scheme. The Even-Mansour construction turns a publicly known, random permutation $P : \{0, 1\}^n \rightarrow \{0, 1\}^n$ into a keyed, pseudorandom permutation

$$E_{k_1, k_2}^P : \{0, 1\}^n \longrightarrow \{0, 1\}^n$$

$$x \longmapsto P(x \oplus k_1) \oplus k_2$$

where $k_1, k_2 \in \{0, 1\}^n$, and \oplus denotes bitwise XOR. This scheme is relevant in two settings:

1. simply as a source of pseudorandomness; in this setting, oracle access to P is provided to all parties.
2. as a block cipher; now oracle access to both P and P^{-1} is provided to all parties. Access to P^{-1} is required for decryption. One can then ask if E^P is a PRP (adversary gets access to E^P), or a strong PRP (adversary gets access to both E^P and its inverse).

In all of these settings, Even-Mansour is known to be (information-theoretically) secure against classical polynomial-time adversaries [7].

Quantum CPA attack on standard scheme. The proofs of classical security of Even-Mansour carry over immediately to the setting of quantum adversaries with only classical access to the relevant oracles. However, if an adversary is granted quantum oracle access to the P and E^P oracles, but no access at all to the inverse oracles, then Even-Mansour is easily broken. This attack was first described in [18]; a complete analysis is given in [13]. The attack is simple: First, one uses the quantum oracles for P and E^P to create a quantum oracle for $P \oplus E^P$, i.e., the function

$$f(x) = P(x) \oplus P(x \oplus k_1) \oplus k_2.$$

One then runs Simon’s algorithm [28] on the function f . The claim is that, with high probability, Simon’s algorithm will output k_1 . To see this, note that f satisfies half of Simon’s promise, namely $f(x \oplus k_1) = f(x)$. Moreover, if it is classically secure, then it *almost* satisfies the entire promise. More precisely, for any fixed P and random pair (x, y) , either the probability of a collision $f(x) = f(y)$ is low enough for Simon’s algorithm to succeed, or there are so many collisions that there exists a classical attack [13]. Once we have recovered k_1 , we also immediately recover k_2 with a classical query, since $k_2 = E_{k_1, k_2}^P(x) \oplus P(x \oplus k_1)$ for any x .

Hidden Shift Even-Mansour. To address the above attack, we propose simple variants of the Even-Mansour scheme. The construction generalizes the standard Even-Mansour scheme in the manner described in Section 1. Each variant is parameterized by a family of exponentially-large finite groups G . The general construction is straightforward to describe. We begin with a public permutation $P : G \rightarrow G$, and from it construct a family of keyed permutations

$$E_{k_1, k_2}^P(x) = P(x \cdot k_1) \cdot k_2,$$

where k_1, k_2 are now uniformly random elements of G , and \cdot denotes composition in G . The formal definition, as a block cipher, follows.

Scheme 1 (Hidden Shift Even-Mansour block cipher). *Let \mathcal{G} be a family of finite, exponentially large groups, satisfying the efficient encoding conditions given in Section 3.2. The scheme consists of three polynomial-time algorithms, parameterized by a permutation P of the elements of a group G in \mathcal{G} :*

- **KeyGen** : $\mathbb{N} \rightarrow G \times G$; on input $|G|$, outputs $(k_1, k_2) \in_R G \times G$;
- **Enc** $_{k_1, k_2}^P$: $G \rightarrow G$; defined by $m \mapsto P(m \cdot k_1) \cdot k_2$;
- **Dec** $_{k_1, k_2}^P$: $G \rightarrow G$; defined by $c \mapsto P^{-1}(c \cdot k_2^{-1}) \cdot k_1^{-1}$.

For simplicity of notation, we set $E_{k_1, k_2}^P := \text{Enc}_{k_1, k_2}^P$. Note that $\text{Dec}_{k_1, k_2}^P = (E_{k_1, k_2}^P)^{-1}$. Correctness of the scheme is immediate; in the next section, we present several arguments for its security in various settings. All of these arguments are based on the conjectured hardness of certain HIDDEN SHIFT problems over \mathcal{G} .

4.2 Security reductions

We consider two settings. In the first, the adversary is given oracle access to the permutation P , and then asked to distinguish the Even-Mansour cipher E_{k_1, k_2}^P from a random permutation unrelated to P . In the second setting, the adversary is given oracle access to P , P^{-1} , as well as E_{k_1, k_2}^P and its inverse; the goal in this case is to recover the key (k_1, k_2) (or some part thereof).

Distinguishability from random. We begin with the first setting described above. We fix a group G , and let \mathcal{P}_G denote the family of all permutations of G . Select a uniformly random $P \in \mathcal{P}_G$. The encryption map for the Hidden Shift Even-Mansour scheme over G can be written as

$$E_{k_1, k_2}^P = L_{k_2} \circ P \circ L_{k_1}.$$

If we have oracle access to P , then this is clearly an efficiently computable subfamily of \mathcal{P}_G , indexed by key-pairs. For pseudorandomness, the relevant problem is then to distinguish E^P from a random permutation which is unrelated to the oracle P .

Problem 6 (Even-Mansour Distinguishability (EMD)). Given oracle access to permutations $P, Q \in \mathcal{P}_G$ and a promise that either (i.) both P and Q are random, or (ii.) P is random and $Q = E_{k_1, k_2}^P$ for random k_1, k_2 , decide which is the case.

It is straightforward to connect this problem to the decisional version of RANDOM HIDDEN SHIFT, as follows.

Proposition 4. *If there exists a QPT \mathcal{D} for EMD on \mathcal{G} , then there exists a QPT for the DRHS problem on \mathcal{G} , with soundness and completeness at most negligibly different from those of \mathcal{D} .*

Proof. Let f, g be the two oracle functions for the DRHS problem over G . We know that f is a random function from G to G , and we must decide if g is also random, or simply a shift of f . We sample t_1, t_2 uniformly at random from G , and provide \mathcal{D} with oracles f (in place of P), and $g' := L_{t_2} \circ g \circ L_{t_1}$ (in place of E^P .) We then simply output what \mathcal{D} outputs. Note that (f, g) are uniformly random permutations if and only if (f, g') are. In addition, $g = f \circ L_s$ if and only if $g' = L_{t_2} \circ f \circ L_{st_1}$. It follows that the input distribution to \mathcal{D} is as in EMD, modulo the fact that the oracles in DRHS are random functions rather than random permutations. The error resulting from this is at most negligible, by the collision-finding bound of Zhandry [30]. \square

Next, we want use quantum-secure pseudorandom functions (qPRFs) to amplify the DRHS distinguisher, and then apply the reduction from HIDDEN SHIFT given in Proposition 3. Combining this with Proposition 4, we arrive at a complete security reduction.

Theorem 2. *Suppose qPRFs exist. Let \mathcal{G} be either the $\mathbb{Z}/2^n$ group family or the S_n group family. Under Assumption 4, the Hidden Shift Even-Mansour cipher over \mathcal{G} is a quantum-secure pseudorandom function.*

Proof. Let \mathcal{G} be either the $\mathbb{Z}/2^n$ group family, or the S_n group family. If the Even-Mansour cipher over \mathcal{G} is not a qPRP, then by Definition 3, there exists an algorithm \mathcal{D}_{EMD} for the EMD problem with total (i.e., completeness plus soundness) error at most $1 - 1/s(n)$ for some polynomial s . To give the adversary as much freedom as possible, we assume that the probability of selecting the

public permutation P is taken into account here; that is, \mathcal{D}_{EMD} need only succeed with inverse-polynomial probability over the choices of permutation P , keys k_1, k_2 , and its internal randomness.

By [Proposition 4](#), we then also have a DRHS algorithm $\mathcal{D}_{\text{DRHS}}$ with error at most $1 - 1/s(n)$ (up to negligible terms.) We can amplify this algorithm by means of a qPRF F . Given functions f, g for the DRHS problem on G , we select a random key k for F and a random group element $t \in G$. We then call $\mathcal{D}_{\text{DRHS}}$ with oracles

$$f_k := F_k \circ f \quad \text{and} \quad g_{k,t} := F_k \circ g \circ L_t$$

Note that, to any efficient quantum algorithm, (i.) f and g are random if and only if f_k and g_{k,t_1,t_2} are, and (ii.) $g(x) = f(sx)$ if and only if $g_{k,t}(x) = t_2 f_k(st_1 x)$. We know that $\mathcal{D}_{\text{DRHS}}$ will succeed with probability $1 - 1/s(n)$, except the probability is now taken over the choice of t and k (rather than f and g .) We repeat this process with different random choices of k and t . A straightforward application of a standard Chernoff bound shows that, after $O(p(n))$ runs, we will correctly distinguish with $1 - \text{negl}(n)$ probability.

Finally, we apply [Proposition 3](#), to get an algorithm for RANDOM HIDDEN SHIFT with negligible error; by [Proposition 1](#), we get an equally strong algorithm for HIDDEN SHIFT. \square

Key recovery attacks. We now consider partial or complete key recovery attacks, in the setting where the adversary also gets oracle access to the inverses of P and E_{k_1, k_2}^P . Note that, for the Even-Mansour cipher on any group G , knowing the first key k_1 suffices to produce the second key k_2 , since

$$k_2 = P(x \cdot k_1)^{-1} E_{k_1, k_2}^P(x)$$

for every $x \in G$.

We remark that giving security reductions is now complicated by the fact that RANDOM HIDDEN SHIFT and its variants all become trivial if we are granted even a partial ability to invert f or g ; querying $f^{-1} \circ g$ on any input x produces $x \cdot s^{-1}$, which immediately yields the shift s . However, we can still give a nontrivial reduction, as follows.

Theorem 3. *Consider the Even-Mansour cipher over $G \times G$, for any group G . Suppose there exists a QPT which, when granted oracle access to P , E_{k_1, k_2}^P , and their inverses, outputs k_1, k_2 . Then there exists an efficient quantum algorithm for the HIDDEN SHIFT problem over G .*

Proof. We use additive notation for convenience; the argument holds generally. Suppose we are given oracle access to functions $f, g : G \rightarrow G$ and a promise that there exists $s \in G$ such that $f(x) = g(x + s)$ for all $x \in G$. We define the following oracles, which can be constructed from access to f and g . First, we have permutations $P_f, P_g : G \times G \rightarrow G \times G$ defined by

$$P_f(x, y) = (x, y + f(x)) \quad \text{and} \quad P_g(x, y) = (x, y + g(x)).$$

Now we sample keys $k_1 = (x_1, y_1), k_2 = (x_2, y_2)$ from $G \times G$ and define the function $E := E_{k_1, k_2}^{P_f}$. To the key-recovery adversary \mathcal{A} for Even-Mansour over $G \times G$, we provide the oracles E and E^{-1} for the encryption/decryption oracles, and the oracles P_g and P_g^{-1} for the public permutation oracles.

To see that we can recover the shift s from the output of \mathcal{A} , we rewrite E in terms of g , as follows:

$$\begin{aligned} E(x, y) &= P_f(x + x_1, y + y_1) + (x_2, y_2) \\ &= (x + x_1, y + y_1 + f(x + x_1)) + (x_2, y_2) \\ &= (x + x_1 + s, y + y_1 + f(x + x_1)) + (x_2 - s, y_2) \\ &= (x + x_1 + s, y + y_1 + g(x + x_1 + s)) + (x_2 - s, y_2) \\ &= P_g(x + x_1 + s, y + y_1) + (x_2 - s, y_2). \end{aligned}$$

After complete key recovery, \mathcal{A} will output $(x_1 + s, y_1)$ and $(x_2 - s, y_2)$, from which we easily deduce s . \square

Remark. The reduction above focuses on the problem of recovering the entire key. Note that for certain groups, e.g., \mathbb{Z}/p for prime p , predicting any bit of the key with inverse-polynomial advantage is sufficient to recover the entire key (see Håstad and Näslund [11]). In such cases we may conclude that predicting individual bits of the key is difficult.

5 Hidden Shift CBC-MACs

5.1 Generalizing the Encrypted-CBC-MAC scheme

The standard scheme. The standard Encrypted-CBC-MAC construction requires a pseudorandom permutation $E_k : \{0, 1\}^n \rightarrow \{0, 1\}^n$. A message m is subdivided into blocks $m = m_1 || m_2 || \dots || m_l$, each of length n . The tag is then computed by repeatedly encrypting-and-XORing the message blocks, terminating with one additional round of encryption with a different key. Specifically, we set

$$\text{CBC-MAC}_{k, k'} := E_{k'}(E_k(m_l \oplus E_k(\dots E_k(m_2 \oplus E_k(m_1)) \dots))).$$

This yields a secure MAC for variable-length messages.

Quantum CPA attack on standard scheme. If we are granted quantum CPA access to $\text{CBC-MAC}_{k, k'}$, then there is a $(\mathbb{Z}/2)^n$ -hidden-shift attack, described below. This attack was described in [13]; another version of the attack appears in [26]. Consider messages consisting of two blocks, and fix the first block to be one of two distinct values $\alpha_0 \neq \alpha_1$. We use the oracle for $\text{CBC-MAC}_{k, k'}$ to construct an oracle for the function

$$f(b, x) := \text{CBC-MAC}_{k, k'}(\alpha_b || x) = E_{k'}(E_k(x \oplus E_k(\alpha_b))).$$

Note that f satisfies Simon’s promise, since

$$f(b \oplus 1, x \oplus E_k(\alpha_0) \oplus E_k(\alpha_1)) = f(b, x)$$

for all b, x . We can thus run Simon’s algorithm to recover the string $s_k = E_k(\alpha_0) \oplus E_k(\alpha_1)$. Knowledge of s_k enables us to find an exponential number of collisions, since

$$\text{CBC-MAC}_{k,k'}(\alpha_0||x) = \text{CBC-MAC}_{k,k'}(\alpha_1||x \oplus E_k(\alpha_0) \oplus E_k(\alpha_1)).$$

In particular, this CBC-MAC does not satisfy the Boneh-Zhandry notion of a secure MAC in the quantum world [2].

Hidden Shift CBC-MAC. We propose generalizing the Encrypted-CBC-MAC construction above, by allowing the bitwise XOR operation to be replaced by composition in some exponentially-large family of finite groups G . Each message block is then identified with an element of G , and we view the pseudorandom permutation E_k as a permutation of the group elements of G . We then define

$$\begin{aligned} \text{CBC-MAC}_{k,k'}^G : G^* &\longrightarrow G \\ (m_1, \dots, m_l) &\longmapsto E_{k'}(E_k(m_l \cdot E_k(\dots E_k(m_2 \cdot E_k(m_1)) \dots))), \end{aligned}$$

where \cdot denotes the group operation in G .

Scheme 2 (Hidden Shift Encrypted-CBC-MAC). *Let G be a family of finite, exponentially large groups satisfying the efficient encoding conditions given in Section 3.2. Let $E_k : G \rightarrow G$ be a quantum-secure pseudorandom permutation. The scheme consists of three polynomial-time algorithms:*

- **KeyGen**; on input $|G|$, outputs two keys k, k' using key generation for E ;
- **Mac** $_{k,k'} : m \mapsto E_{k'}(E_k(m_l \cdot E_k(\dots E_k(m_2 \cdot E_k(m_1)) \dots)))$;
- **Ver** $_{k,k'} : (m, t) \mapsto \text{accept}$ if $\text{Mac}_{k,k'}(m) = t$, and **reject** otherwise.

We consider the security of this scheme in the next section.

5.2 Security reduction

We now give a reduction from the RANDOM HIDDEN SHIFT problem to collision-finding in the above CBC-MAC.

Theorem 4. *Assume q PRFs exist. Let \mathcal{G} be either the $\mathbb{Z}/2^n$ group family or the S_n group family. Under Assumption 4, the Hidden-Shift CBC-MAC over \mathcal{G} is a collision-free function.*

Proof. For simplicity, we assume that the collision-finding adversary finds collisions between equal-length messages. This is of course trivially true, for example, if the MAC is used only for messages of some a priori fixed length.

Suppose we are given an instance of the HIDDEN SHIFT problem, i.e., a pair of functions F_0, F_1 with the promise that F_0 is random and F_1 is a shift of F_0 . We have at our disposal a QPT \mathcal{A} which finds collisions in the Hidden Shift Encrypted-CBC-MAC. We assume without loss of generality that, whenever \mathcal{A} outputs a collision (c, c') , there is no pair of prefixes of (c, c') that also give a valid collision; indeed, we can easily build an \mathcal{A}' which, whenever such prefixes exist, simply outputs the prefix collision instead.

We assume for the moment that the number of message blocks in c and c' is the same number t . Since the number of blocks and the running time of \mathcal{A} are polynomial, we can simply guess t , and we will guess correctly with inverse-polynomial probability. We run \mathcal{A} with a modified oracle \mathcal{O} which “inserts” our hidden shift problem at stage t . This is defined as follows.

Let m be our input message, and l the number of blocks. If $l < t$, we simply output the usual Encrypted-CBC-MAC of m . If $l \geq t$, we first perform $t - 1$ rounds of the CBC procedure, computing a function

$$h(m) := E_k(m_{t-1} \cdot E_k(\dots E_k(m_2 \cdot E_k(m_1)) \dots)).$$

Note that h only depends on the first $t - 1$ blocks of m . Next, we choose a random bit b (e.g., by applying a PRF to m), and compute $F_{b(m)}(m_t \cdot h(m))$. We then finish the rest of the rounds of the CBC procedure, outputting

$$O(m) := E'_k(E_k(m_l \cdot E_k(\dots E_k(F_{b(m)}(m_t \cdot h(m))) \dots)).$$

It's not hard to see that the distribution that the adversary observes will be indistinguishable from the usual Encrypted-CBC-MAC. Suppose a collision (m, m') is output. We set $x_1 = m_1 || m_2 || \dots || m_{t-1}$ and $x_2 = m'_1 || m'_2 || \dots || m'_{t-1}$ and $y_1 = m_t$ and $y_2 = m'_t$. The collision then means that

$$F_{b(m)}(y_1 \cdot h(x_1)) = F_{b(m')}(y_2 \cdot h(x_2)).$$

Since $m \neq m'$, with probability $1/2$ we have $b(m) \neq b(m')$. We repeat \mathcal{A} until we achieve inequality of these bits. We then have

$$F_0(y_1 \cdot h(x_1)) = F_1(y_2 \cdot h(x_2)) = F_0(y_2 \cdot h(x_2) \cdot s)$$

and so the shift is simply $s = y_2^{-1} h(x_2)^{-1} y_1 h(x_1)$. □

6 Thwarting the Simon attack on other schemes

It is reasonable to conjecture that our transformation secures (classically secure) symmetric-key schemes against quantum CPA attacks, generically. So far, we have only been able to give complete security reductions in the cases of the Even-Mansour cipher and the Encrypted-CBC-MAC. For the case of all other schemes vulnerable to the Simon algorithm attacks of [17, 18, 13], we can only say that the attack is thwarted by passing from $(\mathbb{Z}/2)^n$ to $\mathbb{Z}/2^n$ or S_n . We now briefly outline two cases of particular note. For further details, see [Appendix B](#).

The first case is the Feistel network construction, which transforms random functions into pseudorandom permutations. While the three-round Feistel cipher is known to be classically secure [19], no security proof is known in the quantum CPA case, for any number of rounds. In [17], a quantum CPA attack is given for the three-round Feistel cipher, again based on Simon’s algorithm. The attack is based on the observation that, if one fixes the first half of the input to one of two fixed values $\alpha_0 \neq \alpha_1$, then the output contains one of two functions $f_{\alpha_0}, f_{\alpha_1}$, which are $(\mathbb{Z}/2)^n$ -shifts of each other. However, if we instead replace each bitwise XOR in the Feistel construction with addition modulo $\mathbb{Z}/2^n$, the two functions become $\mathbb{Z}/2^n$ -shifts, and the attack now requires a cyclic HIDDEN SHIFT subroutine.

The second case is what [13] refer to as the “quantum slide attack,” which uses Simon’s algorithm to give a linear-time quantum CPA attack, an exponential speedup over classical slide attacks. The attack works against ciphers $E_{k,t}(x) := k \oplus (R_k)^t(x)$ which consist of t rounds of a function $R_k(x) := R(x \oplus k)$. In the attack, one simply observes that $E_{k,t}(R(x))$ is a shift of $R(E_{k,t}(x))$ by the key k , and then applies Simon’s algorithm. To defeat this attack, we simply work over $\mathbb{Z}/2^n$, setting $E_{k,t}(x) := k + (R_k)^t(x)$ and $R_k(x) := R(x + k)$. It’s easy to see that the same attack now requires a HIDDEN SHIFT subroutine for $\mathbb{Z}/2^n$.

Acknowledgments

G.A. would like to thank Tommaso Gagliardoni and Christian Majenz for helpful discussions. G.A. acknowledges financial support from the European Research Council (ERC Grant Agreement 337603), the Danish Council for Independent Research (Sapere Aude) and VILLUM FONDEN via the QMATH Centre of Excellence (Grant 10059). A. R. acknowledges support from NSF grant IIS-1407205.

Bibliography

- [1] Gorjan Alagic, Cristopher Moore, and Alexander Russell. Quantum algorithms for simon’s problem over general groups. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 1217–1224, 2007. URL <http://dl.acm.org/citation.cfm?id=1283383.1283514>.
- [2] Dan Boneh and Mark Zhandry. *Quantum-Secure Message Authentication Codes*, pages 592–608. Springer, Berlin, Heidelberg, 2013. doi:10.1007/978-3-642-38348-9_35.
- [3] Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on post-quantum cryptography. Technical report, National Institute of Standards and Technology, 2016. URL <http://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf>.
- [4] Andrew M. Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *J. Mathematical Cryptology*, 8(1):1–29, 2014. doi:10.1515/jmc-2012-0016.
- [5] Hang Dinh, Cristopher Moore, and Alexander Russell. Limitations of single coset states and quantum algorithms for code equivalence. *Quantum Info. Comput.*, 15(3-4):260–294, March 2015.
- [6] Mark Ettinger, Peter Høyer, and Emanuel Knill. The quantum query complexity of the hidden subgroup problem is polynomial. *Information Processing Letters*, 91(1):43–48, 2004. ISSN 0020-0190. doi:10.1016/j.ipl.2004.01.024.
- [7] Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *Journal of Cryptology*, 10(3):151–161, 1997. ISSN 1432-1378. doi:10.1007/s001459900025.
- [8] Stephen Fenner and Yong Zhang. On the complexity of the hidden subgroup problem. *International Journal of Foundations of Computer Science*, 24(8):1221–1234, 2013.
- [9] Katalin Friedl, Gábor Ivanyos, Frédéric Magniez, Miklos Santha, and Pranab Sen. Hidden translation and translating coset in quantum computing. *SIAM J. Comput.*, 43(1):1–24, 2014. doi:10.1137/130907203. URL <http://dx.doi.org/10.1137/130907203>.
- [10] M. L. Furst, J. E. Hopcroft, and E. M. Luks. Polynomial-time algorithms for permutation groups. In *Proceedings of the 21st IEEE Symposium on Foundations of Computer Science*, 1980.
- [11] J. Håstad and M. Näslund. The security of individual RSA bits. In *Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on*, pages 510–519, Nov 1998. doi:10.1109/SFCS.1998.743502.
- [12] Sean Hallgren, Cristopher Moore, Martin Rötteler, Alexander Russell, and Pranab Sen. Limitations of quantum coset states for graph isomorphism. *J. ACM*, 57(6):34:1–34:33, November 2010. doi:10.1145/1857914.1857918.

- [13] M. Kaplan, G. Leurent, A. Leverrier, and M. Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. Technical Report quant-ph/1602.05973, arXiv, February 2016.
- [14] A. Y. Kitaev. Quantum measurements and the abelian stabilizer problem. Technical Report quant-ph/9511026, arXiv, November 1995.
- [15] G. Kuperberg. Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. Technical Report quant-ph/1112.3333, arXiv, December 2011.
- [16] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM J. Comput.*, 35(1):170–188, July 2005. doi:10.1137/S0097539703436345.
- [17] H. Kuwakado and M. Morii. Quantum distinguisher between the 3-round feistel cipher and the random permutation. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 2682–2685, June 2010. doi:10.1109/ISIT.2010.5513654.
- [18] Hidenori Kuwakado and Masakatu Morii. Security on the quantum-type Even-Mansour cipher. In *Proceedings of the International Symposium on Information Theory and its Applications (ISITA)*, pages 312–316. IEEE Computer Society, 2012. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=6400943.
- [19] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):337–386, 1988.
- [20] Cristopher Moore, Alexander Russell, and Piotr Śniady. On the impossibility of a quantum sieve algorithm for graph isomorphism. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 536–545, New York, NY, USA, 2007. ACM. doi:10.1145/1250790.1250868.
- [21] Cristopher Moore, Alexander Russell, and Leonard J. Schulman. The symmetric group defies strong Fourier sampling. *SIAM J. Comput.*, 37:1842–1864, March 2008. doi:10.1137/050644896.
- [22] Sarvar Patel, Zulfikar Ramzan, and Ganpathy S. Sundaram. Luby-rackoff ciphers: Why xor is not so exclusive. In *Revised Papers from the 9th Annual International Workshop on Selected Areas in Cryptography*, SAC '02, pages 271–290, London, UK, UK, 2003. Springer-Verlag. ISBN 3-540-00622-2. URL <http://dl.acm.org/citation.cfm?id=646558.694919>.
- [23] Oded Regev. A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space. Technical Report quant-ph/0406151, arXiv, June 2004.
- [24] Oded Regev. Quantum computation and lattice problems. *SIAM J. Comput.*, 33(3):738–760, March 2004. doi:10.1137/S0097539703440678.
- [25] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM. doi:10.1145/1060590.1060603.

- [26] T. Santoli and C. Schaffner. Using Simon’s algorithm to attack symmetric-key cryptographic primitives. Technical Report quant-ph/1603.07856, arXiv, March 2016.
- [27] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *FOCS 1994*, pages 124–134. IEEE Computer Society Press, 1994.
- [28] Daniel R. Simon. On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, October 1997. doi:10.1137/S0097539796298637.
- [29] Mark Zhandry. How to construct quantum random functions. In *FOCS 2012*, pages 679–687. IEEE, 2012.
- [30] Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Info. Comput.*, 15(7-8):557–567, May 2015. ISSN 1533-7146. URL <http://dl.acm.org/citation.cfm?id=2871411.2871413>.

A Hidden Subgroups and Hidden Shifts

Basic definitions. We now briefly discuss the HIDDEN SUBGROUP PROBLEM, which is closely related to HIDDEN SHIFT.

Problem 7 (HIDDEN SUBGROUP PROBLEM (HSP)). Let G be a group and S a set. Given a function $f : G \rightarrow S$, and a promise that there exists $H \leq G$ such that f is constant and distinct on the right cosets of H , output a complete set of generators for H .

Another, equivalent formulation of the HSP promise on f is that for $x \neq y$, $f(x) = f(y)$ iff $x = h \cdot y$ for $h \in H$. As before, one can also consider decision versions of HSP (e.g., where one has to decide if f hides a trivial or nontrivial subgroup) and promise versions where the function f is a random function satisfying the constraint that $f(hx) = x$ for all $x \in G$ and $h \in H$. This last variant is important for our purposes so we separately define it.

Problem 8 (RANDOM HIDDEN SUBGROUP PROBLEM (RHSP)). Let G be a group and S a set. Given a function $f : G \rightarrow S$ chosen uniformly among all functions for which $f(x) = f(hx)$ for all $x \in G$ and $h \in H$, output a complete set of generators for H .

Some reductions Traditionally, the Hidden Subgroup problem (HSP) has played a prominent role in the literature, as it offers a simple framework to which many other problems can be directly reduced. Indeed, there is a general reduction from HS to HSP.

A canonical reduction from HS to HSP. Consider an instance of a HIDDEN SHIFT problem over G given by the functions $f_0, f_1 : G \rightarrow S$ such that $f_0(x) = f_1(x \cdot s)$. Recall that the *wreath product* $K = G \wr \mathbb{Z}/2$ is the semi-direct product

$(G \times G) \rtimes \mathbb{Z}/2$, where the action of the nontrivial element of $\mathbb{Z}/2$ on $G \times G$ is the swap $(a, b) \mapsto (b, a)$. Now define the function

$$\begin{aligned} \varphi : (G \times G) \rtimes \mathbb{Z}/2 &\longrightarrow S \times S \\ ((x, y), b) &\longmapsto (f_b(x), f_{b \oplus 1}(y)). \end{aligned}$$

One then easily checks that the function φ is constant and distinct on the cosets of the order-two subgroup of K generated by $((s, s^{-1}), 1)$.

We remark that the reduction above can significantly “complicate” the underlying group. In particular, note that $A \rtimes \mathbb{Z}/2$ is always non-abelian (unless the action of $1 \in \mathbb{Z}/2 = \{0, 1\}$ on A is trivial). Note that this reduction does not yield a reduction from RHS to RHSP as the resulting HSP instance is not uniformly random in the fully random case.

The special case of $(\mathbb{Z}/2)^n$; reductions from RHS to RHSP. On $(\mathbb{Z}/2)^n$, the HIDDEN SHIFT problem can be reduced to the RANDOM HIDDEN SUBGROUP PROBLEM on $(\mathbb{Z}/2)^n$ via a special reduction that exploits $\mathbb{Z}/2$ structure. Specifically, for a pair of injective functions $f_0, f_1 : (\mathbb{Z}/2)^n \rightarrow S$ (for which $f_1(x) = f_2(x \oplus s)$), construct the function $f : (\mathbb{Z}/2)^{n+1} \rightarrow S$ so that

$$g(bx) = f_b(x), \quad \text{for } b \in \mathbb{Z}/2 \text{ and } x \in (\mathbb{Z}/2)^n.$$

Then observe that g hides the subgroup generated by $1s$.

Note, furthermore, that if the f_i are (independent) random functions, then the same can be said of g ; likewise, if f_1 is a shift of the random function f_0 , the function g is precisely a random function subject to the constraint that $g(x) = g(x \oplus 1s)$; thus this reduces RHS to RHSP. In this RHS setting, it is possible to develop an alternate reduction that more closely resembles the attacks we discussed above. Specifically, given the functions $f_0, f_1 : (\mathbb{Z}/2)^n \rightarrow (\mathbb{Z}/2)^n$ (so that $S = (\mathbb{Z}/2)^n$), consider the oracle $g = f_0 \oplus f_1$. When $f_1(x) = f_0(x \oplus s)$, note that this oracle satisfies the symmetry condition

$$\begin{aligned} [f_0 \oplus f_1](x \oplus s) &= f_0(x \oplus s) \oplus f_1(x \oplus s) \\ &= f_0(x \oplus s \oplus s) + f_1(x) = f_0(x) + f_1(x) = [f_0 \oplus f_1](x) \end{aligned}$$

so that $g = f_0 \oplus f_1$ is a random function subject to the constraint that $g(x) = g(x \oplus s)$, as desired. If the functions f_i are independent, the function g has the uniform distribution. Thus this reduces RHS to RHSP.

B Other Hidden Shift Constructions

B.1 Feistel ciphers

The standard scheme. The Feistel cipher is a method for turning random functions into pseudorandom permutations. The core ingredient is a one-round Feistel cipher, which, for a function $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$, is given by

$$\begin{aligned} F_f : \{0, 1\}^{2n} &\longrightarrow \{0, 1\}^{2n} \\ x || y &\longmapsto y \oplus f(x) || x. \end{aligned}$$

The function f is called the “round function.” The multi-round version of the Feistel cipher is defined by concatenating multiple one-round ciphers, each with a different choice of round function. Of particular interest is the three-round cipher, defined by

$$F_{R_1, R_2, R_3}(x||y) = F_{R_3}(F_{R_2}(F_{R_1}(x||y))).$$

A well-known result of Luby and Rackoff says that, if the R_j are random and independent, then F_{R_1, R_2, R_3} is indistinguishable from a random permutation [19].

Quantum CPA attack on standard scheme. Suppose we are given quantum oracle access to a function F , and promised that F is either a random permutation, or that $F := F_{R_1, R_2, R_3}$ for some unknown, random functions R_j . The following attack was first shown in [17]; a thorough analysis appears in [13]. We first fix two n -bit strings $\alpha_0 \neq \alpha_1$. We then use the oracle for F to build oracles f_0, f_1 defined by

$$f_b(y) := F(\alpha_b||y)|_{n+1}^{2n} \oplus \alpha_b.$$

Here $s|_j^k := s_j s_{j+1} \cdots s_k$. We then run Simon’s algorithm to see if there’s a shift between f_0 and f_1 . If a shift is produced, we output “Feistel.” Otherwise we output “random.”

To see why the attack is successful, first note that if F is a random permutation, then the f_b are random functions. On the other hand, if $F = F_{R_1, R_2, R_3}$, then one easily checks that $f_b(y) = R_2(y \oplus R_1(\alpha_b))$. We then have

$$f_1(y) = f_0(y \oplus (R_1(\alpha_0) \oplus R_1(\alpha_1)))$$

for all y . Since the R_j are random, one can check that there are not too many other collisions [13]. It follows that Simon’s algorithm will output $R_1(\alpha_0) \oplus R_1(\alpha_1)$ with high probability.

Hidden Shift Feistel cipher. For simplicity, we will work over the group $\mathbb{Z}/2^n$. Our construction generalizes to other group families in a straightforward way. Given a function $f : \mathbb{Z}/2^n \rightarrow \mathbb{Z}/2^n$, we define the one-round Feistel cipher using round function f to be

$$\begin{aligned} F_f : \mathbb{Z}/2^n \times \mathbb{Z}/2^n &\longrightarrow \mathbb{Z}/2^n \times \mathbb{Z}/2^n \\ (x, y) &\longmapsto (y + f(x), x). \end{aligned}$$

Since $\mathbb{Z}/2^n \times \mathbb{Z}/2^n \cong \mathbb{Z}/2^{2n}$, we can then view F_f as a permutation on $2n$ -bit strings. For multi-round ciphers, we define composition as before. In particular, given three functions R_1, R_2, R_3 on G , we get the three-round Feistel cipher

$$\begin{aligned} F_{R_1, R_2, R_3} : \mathbb{Z}/2^n \times \mathbb{Z}/2^n &\longrightarrow \mathbb{Z}/2^n \times \mathbb{Z}/2^n \\ (x, y) &\longmapsto F_{R_3}(F_{R_2}(F_{R_1}(x, y))). \end{aligned}$$

Next, we check that the attack of [17] now appears to require a subroutine for the Hidden Shift problem over $\mathbb{Z}/2^n$, contrary to our Cyclic Hidden Shift

Assumption (Assumption 2). We are given a function F on $\mathbb{Z}/2^{2n}$ with the promise that F is either random or a Feistel cipher F_{R_1, R_2, R_3} as above. Proceeding precisely as before, we pick two elements $\alpha_0 \neq \alpha_1$ of $\mathbb{Z}/2^n$ and build two functions

$$f_b(y) := F(\alpha_b, y)|_{n+1}^{2n} - \alpha_b.$$

If F is random, then clearly so are f_0 and f_1 . But if $F = F_{R_1, R_2, R_3}$ then one easily checks that

$$f_b(y) = R_2(y + R_1(\alpha_b)),$$

from which it follows that

$$f_1(y) = R_2(y + R_1(\alpha_1)) = R_2(y + R_1(\alpha_0) - R_1(\alpha_0) + R_1(\alpha_1)) = f_0(y + s)$$

where we set $s := R_1(\alpha_1) - R_1(\alpha_0)$. We are thus presented with a HIDDEN SHIFT problem over the group $\mathbb{Z}/2^n$, which is hard according to Assumption 2. The only known subroutine (analogous to Simon) that one could apply here would be Kuperberg’s algorithm, which would find s in time $2^{\Theta(\sqrt{n})}$ [16]. Defining the Feistel network over other groups (such as S_n) would frustrate all nontrivial quantum-algorithmic approaches, including the Kuperberg approach [20].

B.2 Protecting against quantum slide attacks

Quantum slide attack. Classically, slide attacks are a class of subexponential-time attacks against ciphers which encrypt simply by repeatedly applying some function R_k , with a fixed key k . Kaplan et al. [13] showed how Simon’s algorithm can be used to give a polynomial-time “quantum slide attack” against ciphers of the form

$$E_{k,t} := k \oplus R_k^t(x) = k \oplus (R_k \circ R_k \cdots \circ R_k)(x),$$

where $R_k(x) = R(x \oplus k)$, and R is a known permutation. As usual, the attack requires quantum CPA access to $E_{k,t}$. The attack follows directly from the observation that the functions

$$f_b(x) = \begin{cases} E_{k,t}(R(x)) \oplus x & \text{if } b = 0, \\ R(E_{k,t}(x)) \oplus x & \text{if } b = 1. \end{cases}$$

are shifts of each other by the key k , i.e., $f_0(x \oplus k) = f_1(x)$ for all x . This means we can extract k with Simon’s algorithm.

Eliminating the attack via hidden shifts. Following our established pattern, we adapt schemes $E_{k,t}$ to use modular addition over $\mathbb{Z}/2^n$ instead of bitwise XOR. Given a permutation R of $\{0, 1\}^n$, we now set

$$R_k(x) := R(x + k) \quad \text{and} \quad E_{k,t} := k + R_k^t(x).$$

Proceeding with the attack as before, we now define

$$f_b(x) = \begin{cases} E_{k,t}(R(x)) - x & \text{if } b = 0, \\ R(E_{k,t}(x)) - x & \text{if } b = 1. \end{cases}$$

We then check that

$$\begin{aligned} f_0(x+k) &= E_{k,t}(R(x+k)) - (x+k) \\ &= k + R_k^t(R(x+k)) - x - k \\ &= R(E_{k,t}(x)) - x \\ &= f_1(x). \end{aligned}$$

Continuing as in the Simon attack would now require a solution to the HIDDEN SHIFT problem over $\mathbb{Z}/2^n$.