Block Cipher Doubling for a Post-Quantum World

Maria Naya Plasencia, Ritam Bhaumik, André Chailloux, Paul Frixons, Bart Mennink

Block Cipher Doubling for a Post-Quantum World *

Ritam Bhaumik[1,2] André Chailloux[1] Paul Frixons[1,3,4]
Bart Mennink[5] María Naya-Plasencia[1]

[1] Inria, Paris, France
[2] EPFL, Switzerland
[3] Orange Labs, Paris, France
[4] Loria, Nancy, France
[5] Radboud University, Nijmegen, The Netherlands
ritam.bhaumik@epfl.ch andre.chailloux@inria.fr
paul.frixons@inria.fr b.mennink@cs.ru.nl
maria.naya_plasencia@inria.fr

Abstract. In order to maintain a similar security level in a post-quantum setting, many symmetric primitives should have to double their keys and increase their state sizes. So far, no generic way for doing this is known that would provide convincing quantum security guarantees. In this paper we propose a new generic construction, QuEME, that allows to double the key and the state size of a block cipher. The QuEME design is inspired by the ECB-Mix-ECB (EME) construction, but is defined for a different choice of mixing function that withstands our new quantum superposition attack that exhibits a periodic property found in collisions and that breaks EME and a large class of variants of it. We prove that QuEME achieves n-bit security in the classical setting, where n is the block size of the underlying block cipher, and at least $n/6$-bit security in the quantum setting. We propose a concrete instantiation of this construction, called Double-AES, that is built with variants of AES-128.

Keywords: block cipher, length doubler, post-quantum security, superposition attacks, security proofs, Double-AES, cryptanalysis.

1 Introduction

For a long time, it was accepted that symmetric primitives only needed to double their key length in order to stay resistant to quantum attackers. Although new attacks in powerful models [38, 41, 42] have shown that a more in-depth study is needed and that some particular scenarios are dangerous, for the majority

* This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 714294 - acronym QUASYModo). Bart Mennink was supported by the Netherlands Organisation for Scientific Research (NWO) under grant VI.Vidi.203.099.

of symmetric primitives, the best quantum attacks achieve at most a square-root speed-up compared to the classical attack. Consequently, most of these attacks would indeed be infeasible with a double-sized key. Nevertheless, no generic, simple, and efficient way is known for doubling the key size of a primitive, and the best known candidate for this purpose—the FX construction [40]—was proven to be insecure with respect to quantum attacks in the superposition model [44], though it has been shown to fare better in weaker models [37]. Other key-extension modes like the two-key Even Mansour [1] could also be shown secure only in weaker models.

Additionally, Chailloux et al. [12], demonstrated that attacks on modes exploiting internal collisions, i.e., that depend on the internal size of the primitives, might also render the primitives weaker against quantum adversaries. In these cases, doubling the key length might not be enough, and also the internal size of the primitive should also be increased. A new post-quantum symmetric family of primitives—Saturnin [11]—was proposed to address this concern. The block cipher which forms the core of this family has a state size of 256 bits, allowing much more reasonable security claims regarding all types of quantum attacks.

A generic and provably secure way of extending secure classical constructions into new ones with doubled key as well as doubled state size is still an interesting question that has been widely studied without a satisfactory answer being found so far. Hosoyamada and Iwata [32] proved that the 4-round Luby-Rackoff construction (LR4) is a qPRP. However, in order to build secure post-quantum constructions, we need also to take into account the decryption direction, and Ito et al. [35] showed that LR4 has a quantum attack when we allow both encryption and decryption queries. A natural candidate for resisting this attack would be LR5, i.e., 5-round Luby-Rackoff. Unfortunately, with the proof techniques available at present, proving the quantum security of LR5 is very challenging. It is not possible to use the same database technique as in the proof of Hosoyamada and Iwata [32], since there is no known way yet of generalizing database oracles to permutations, and the equations governing the internal variables are quite complex with many variables, making ad-hoc techniques difficult to apply. Moreover, LR5 could achieve $(k/2)\log n$ bits of security at best, where n is the size of the input to the round function, and k the size of one round key, in light of the quantum attack of Dong and Wang [21].

In this work, we aim to solve the problem of designing a novel symmetric cryptographic mode with doubled key and doubled state size, in such a way that its post-quantum security remains comparable to the original classical one. More detailed, we target the design of a construction that provides ***n*-bit security**, where n is the block size of the underlying block cipher, both in the classical and quantum setting. In other words, we aim to design a scheme providing the same resistance to all attacks as an ideal block cipher with $2n$-bit block and key size would provide against all possible quantum adversaries, and that will be less challenging to prove in the quantum setting.

1.1 Encrypt-Mix-Encrypt and Generalizations in Quantum Setting

The starting point of our quest is the Encrypt-Mix-Encrypt construction. In this construction, one starts with an encryption layer, followed by a mixing layer, and then followed by another encryption layer. The two encryption layers could be ECB based on an n-bit block cipher, and the mixing layer can be based on that block cipher as well. A notable construction following this design is the ECB-Mix-ECB (EME) construction of Halevi and Rogaway [31] (see Figure 1a for the construction on $2n$-bit data blocks): it is a highly parallelizable mode that in its general form extends the domain of a block cipher to arbitrary lengths. ECB layers above and below make it a suitable candidate for resisting quantum attacks, since most Simon-like attacks rely on some part of the input passing through only one block cipher call or being XOR-ed directly to the state, and the ECB layers ensure that every part of the input passes through at least two block cipher calls during both the encryption and decryption routines. We take this general construction, with arbitrary mixing layer, as starting point in Section 3.

However, we show in this paper that the quantum security of this construction is *highly dependent* on the choice of mixing layer. In particular, for our simplified setting of a mode on $2n$-bit data blocks, if we instantiate the mixing layer with any evaluation of a keyed block cipher (such as in the case of EME of Figure 1a [31]), we propose a quantum superposition attack that recovers any of the keys used in the ECB layers in around $2^{n/2}$ quantum time. The attack is described in Section 3.2, and is in fact original and exhibits a periodic property found in collisions. In more detail, our attack uses the uniform superposition of collisions to tailor the target scheme to a simpler to analyze function, exposing a period only for the correct key.

1.2 QuEME: Proposal for Quantum Secure Doubler

The general impossibility result on ECB-Mix-ECB of Section 3.2 demonstrates that one either must make a layer based on more than one primitive evaluation, making it more expensive, or a layer using a compressing primitive. For this, one can use the underlying block cipher where both the data and the key input depend on the outputs of the first ECB layer. However, also this layer construction must be done with care, but we eventually found a mixing layer that works, leading to our new construction QuEME. The construction is depicted in Figure 1b and formally described in Section 4. In a nutshell, the mixing layer adds both data paths coming from the first ECB layer (basically having this layer functioning as a sum of permutations and thus yielding uniform random outputs [5, 19, 47, 52, 54]) and uses it as key input to the block cipher. One of the data paths will be transformed using the block cipher and the output will be added to the other data path (this way, the mixing layer is invertible).

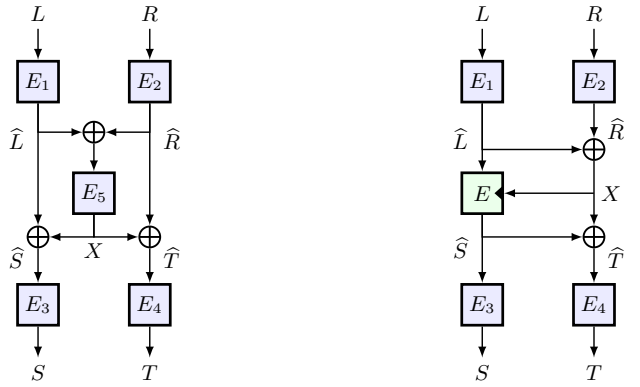


Fig. 1: The EME construction (Figure 1a) and the QuEME construction (Figure 1b). Here, $E_1, \ldots, E_5$ denote five secretly keyed block ciphers and E that takes its key input from the right.

1.3 Classical Security

A first step in our analysis of QuEME is to actually analyze its security in the classical setting. We derive two security proofs. First, we prove in Section 6 security of the construction up to $2^{2n/3}$ evaluations. The proof is modular and gives a good initial impression of the security of QuEME. In addition, it gives an exposition of a new security property of the underlying primitive that we employ, namely random access SPRP security (see Section 2.4). However, the bound is not tight. In Section 5 we derive a generic attack in 2^n queries, that in a nutshell exhausts the entropy of the keyed block ciphers in the encryption layers and relies on the observation that for all these evaluations $(L, R) \mapsto (S, T)$,

$$E_1(L) \oplus E_2(R) = E_3^{-1}(S) \oplus E_4^{-1}(T)$$

(see Figure 1b). (Note: this relation also holds for EME and other similar constructions, which means that our attack also works for these.) Inspired by this, in Section 7 we derive a new security proof up to 2^n evaluations. This proof combines the ideas of the $2^{2n/3}$ security proof and a variant of the mirror theory.

The mirror theory itself has a quite long history [16,52,55], but only recently, a proof of its main variant has been given [15]. In our work, however, we will employ a slightly different variant (see Section 7.1). In support of this variant, we present low-scale computer simulations that confirm it (see Supplementary Material D). These simulations, although restrictive, could be of interest not only for our variant but also for the main variant of the mirror theory.

1.4 Quantum Security

The next step is a quantum analysis of our construction. In Section 8 we show that QuEME achieves at least $n/6$ bits of quantum PRP security. In order to prove this bound, we exploit the fact that the construction starts with two encryption layers and relate the quantum security to the classical security using Zhandry's quantum lower bounds on small range functions. This bound is on par with, for example, the quantum security of LR4 [32] and it shows in particular that there is no collapse in the quantum security as can happen in certain other constructions like LR3 [41]. However, we are not aware of any attack that would perform better than the classical distinguisher (of Section 5) that operates in 2^n evaluations. It is reasonable to believe that our bound is not tight, and in Section 8.2 we discuss potential improvements of our results. A natural way of doing so may be to use Zhandry's technique of recording quantum queries [64] using random permutations instead of functions but this is notoriously hard and arguably not mature enough. The security of QuEME, and its comparison with EME and LR5, is summarized in Table 1.

Table 1: Comparison of the generic security of different extension construction using a block cipher of block size n and key size k, with $k = n$. We note that AES-256, with a state $n = k/2$, provides a much worse level of security when used in modes and when considering attacks on the size of the state (up to $2^{42.6}$ quantumly).

Construction	Classical bound	Quantum bound (Q2)	Classical attack	Quantum attack (Q2)	Expected security
EME	$2^{n/2}$ [31]	—	2^n (Sec. 5)	$2^{n/2}$ (Sec. 3.2)	$2^{n/2}$
LR5	2^n [50]	—	2^n [50]	$2^{n/2}$ [21]	$2^{n/2}$
QuEME	2^n (Sec. 7)	$2^{n/6}$ (Sec. 8)	2^n (Sec. 5)	2^n (Sec. 5)	2^n

1.5 Heuristic Instantiation: Double-AES

Finally, in Section 9, we propose some concrete instantiations of our construction when using (reduced-round versions of) AES-128 as building block. Concretely, we propose Double-AES, where the blocks are slightly tweaked versions (constant-wise) of the full 10-round AES. We additionally propose Double-AES-7, where the number of rounds is reduced to 7 in all blocks, and Double-AES-5-MC, a variant with 5 rounds but that includes the last MC transformation in E_1, E_2, and E. We provide a preliminary cryptanalysis (in Section 9.3) that supports our security claim of n-bit security, and an estimated implementation evaluation (in Section 9.4). The security claim is, for the first time to the best of our knowledge, unified, as we claim a unique security level against all adversaries, regardless of whether they are classical or quantum.

1.6 Outline

We describe the needed notation and security models in Section 2. This section includes our new random access PRP security in Section 2.4. The general Encrypt-Mix-Encrypt setting and its limitations in the quantum setting are discussed in Section 3. Our new construction QuEME is formally described in Section 4. A generic attack in 2^n queries is given in Section 5, a security proof up to $2^{2n/3}$ queries in Section 6, and up to 2^n using a variant of the mirror theory in Section 7. Quantum security of QuEME is given in Section 8. We describe our concrete instantiation Double-AES, including preliminary cryptanalysis and an implementation evaluation, in Section 9. The work is concluded in Section 10.

2 Notation

2.1 Common Definitions and Notations

For $m, n \in \mathbb{N}$, the set of m-to-n-bit functions is denoted $\mathrm{func}(m, n)$ and the set of n-bit permutations indexed by an m-bit key is denoted $\mathrm{perm}(m, n)$. For $m = 0$, i.e., for the set of n-bit permutations, we simply write $\mathrm{perm}(n)$. For a finite set $\mathcal{A}$, we denote by $A \xleftarrow{\$} \mathcal{A}$ the uniform random drawing of A from $\mathcal{A}$. For $m \leq n$, we will write $[m..n]$ to denote the range $\{m, \ldots, n\}$, and $[n] = [1..n]$. We will use the Pochhammer falling factorial power notation

$$(n)_m := n(n-1) \cdot \ldots \cdot (n-m+1).$$

2.2 Distinguishers and Distinguishing Advantage

An adversary $\mathcal{A}$ is an algorithm that gets access to a randomized oracle $\mathcal{O}$ and outputs a decision bit $b \in \{0, 1\}$. We denote this as $\mathcal{A}^{\mathcal{O}}(\cdot) = b$. If the adversarial goal is to distinguish two different randomized oracles $\mathcal{O}$ and $\mathcal{P}$, we denote its advantage as

$$\mathbf{Adv}_{\mathcal{O};\mathcal{P}}(\mathcal{A}) = \left|\Pr\left[\mathcal{A}^{\mathcal{O}} = 1\right] - \Pr\left[\mathcal{A}^{\mathcal{P}} = 1\right]\right|.$$

The oracles $\mathcal{O}$ and $\mathcal{P}$ may be quantum oracles, the adversary $\mathcal{A}$ may be a quantum distinguisher.

2.3 H-Coefficient Technique

Suppose an adversary $\mathcal{A}$ aims to distinguish two oracles $\mathcal{O}$ and $\mathcal{P}$. If we consider $\mathcal{A}$ to be information-theoretic, meaning that its complexity is only measured by the number of oracle calls it makes, we can without loss of generality assume that it is deterministic. We can then use the H-coefficient technique [53] to bound the distance.

Assume we store the entire interaction that $\mathcal{A}$ has with its oracle by a *transcript* τ. Denote by $\mathcal{D}_{\mathcal{O}}$ the probability distribution of transcripts that can be

obtained while interacting with $\mathcal{O}$, and by $\mathcal{D}_{\mathcal{P}}$ the probability distribution of transcripts coming from interaction with $\mathcal{P}$. We say that a transcript is *attainable* if $\Pr[\mathcal{D}_{\mathcal{P}} = \tau] > 0$. Denote by $\mathcal{T}$ the set of all attainable transcripts. The H-coefficient technique states the following about the distinguishing advantage of $\mathcal{A}$.

Lemma 1 (H-coefficient technique [53]). *Consider any partition of attainable transcripts $\mathcal{T}$ into good transcripts $\mathcal{T}_{\text{good}}$ and bad transcripts $\mathcal{T}_{\text{bad}}$. Let ε be such that for all $\tau \in \mathcal{T}_{\text{good}}$,*

$$\frac{\Pr[\mathcal{D}_{\mathcal{O}} = \tau]}{\Pr[\mathcal{D}_{\mathcal{P}} = \tau]} \geq 1 - \varepsilon. \tag{1}$$

Then, for any fixed information-theoretic deterministic adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{O};\mathcal{P}}(\mathcal{A}) \leq \varepsilon + \Pr[\mathcal{D}_{\mathcal{P}} \in \mathcal{T}_{\text{bad}}]$.

A nice and compact proof of the technique can be found in [13, 14]. The H-coefficient technique allows us to partition the set of all attainable transcripts $\mathcal{T}$ wisely so that both ε and $\Pr[\mathcal{D}_{\mathcal{P}} \in \mathcal{T}_{\text{bad}}]$ are small.

The transcripts τ themselves typically simply store the entire interaction that $\mathcal{A}$ has with its oracle, i.e., its query-response tuples. Sometimes, in security proofs it is convenient to consider *extended* transcripts. In this setting, the adversary is given additional information, denoted τ^* generated by a sample $\mathcal{S}$, typically at the end of the security game but before $\mathcal{A}$ outputs its decision bit. Lemma 1 also applies to the setting of extended transcripts.

2.4 Pseudorandom Permutations in Quantum Setting

We follow the well-established notions of (quantum) (S)PRPs [32, 33, 63]. In addition, we will use a variant that we call random access (quantum) (S)PRPs.

2.4.1 (Quantum) PRPs. Let $E : \{0,1\}^k \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be a block cipher. We denote its (quantum) pseudorandom permutation, or (q)PRP, security against an adversary $\mathcal{A}$ as

$$\mathbf{Adv}_E^{\text{(q-)prp}}(\mathcal{A}) = \mathbf{Adv}_{E_K;\Pi}(\mathcal{A}),$$

where $K \xleftarrow{\$} \{0,1\}^k$ and $\Pi \xleftarrow{\$} \text{perm}(n)$. The "$q$-" in the superscript denotes that we consider the quantum setting (where both E_K/Π and P are quantum oracles). The adversary $\mathcal{A}$ may be bounded by a certain number of oracle queries q and time t.

For the classical setting, we will also consider strong PRP, or SPRP, security, where the adversary has both forward and inverse access to its primitive:

$$\mathbf{Adv}_E^{\text{sprp}}(\mathcal{A}) = \mathbf{Adv}_{E_K^{\pm};\Pi^{\pm}}(\mathcal{A}),$$

where $K \xleftarrow{\$} \{0,1\}^k$ and $\Pi \xleftarrow{\$} \text{perm}(n)$. The "$\pm$" in the superscript denotes that we consider two-sided access

2.4.2 Random Access (Quantum) PRPs. Let $E : \{0,1\}^k \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be a block cipher. We will also require a different type of block cipher security, which we dub random access (quantum) (S)PRP, or ra-(q)(S)PRP, security. The idea of ra-(q)(S)PRP security is that the attacker has freedom in the selection of *both* the data and key input to E, *but only in a restricted fashion.* The definition is, admittedly, tailored towards the use of E in our mode, but on the upside, it prevents us from resorting to the ideal cipher model that would be a too strong assumption.

Formally, ra-SPRP security of E against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}_E^{\text{ra-sprp}}(\mathcal{A}) = \mathbf{Adv}_{\mathcal{RA}[\boldsymbol{p},E],\mathcal{RA}[\boldsymbol{p},E]^{-1};\mathcal{RA}[\boldsymbol{p},\tilde{\Pi}],\mathcal{RA}[\boldsymbol{p},\tilde{\Pi}]^{-1}}(\mathcal{A}),$$

where $\boldsymbol{p} = (p_1, \ldots, p_4) \xleftarrow{\$} \text{perm}(n)^4$ and $\tilde{\Pi} \xleftarrow{\$} \text{perm}(n,n)$, and the oracle $\mathcal{RA}[\boldsymbol{p}, F]$ for $F \in \{E, \tilde{\Pi}\}$ operates as follows: on a forward query (A, B) it outputs $(K, X, Y) = (p_1(A) \oplus p_2(B), p_1(A), F(K, X))$, and on an inverse query (A, B) it outputs $(K, X, Y) = (p_3(A) \oplus p_4(B), p_3(A), F^{-1}(K, Y))$. The adversary is only allowed to make offline evaluations of E *before* making its queries. This is justified by the fact that, in our use case, we will use ra-SPRP in a non-adaptive setting anyway (in fact the adversary will never even learn the outputs). There may be an issue if the adversary makes accidentally colliding forward and inverse evaluations, but this issue is captured in the security reduction. Naturally, ra-PRP is defined for adversaries that only have forward access to the oracle and ra-qPRP security for adversaries if we consider the quantum setting. However, even in the quantum setting, we only consider classical queries to $\mathcal{RA}[\boldsymbol{p}, F]$ and its inverse which will be enough for our proof.

We remark that in the ideal cipher model, i.e., if $E \xleftarrow{\$} \text{perm}(n,n)$, the time complexity gets replaced by q' offline queries and one can prove that $\mathbf{Adv}_E^{\text{ra-sprp}}(\mathcal{A}) = \frac{qq'}{2^{2n}}$. In the quantum setting, we can prove $\mathbf{Adv}_E^{\text{ra-q-prp}}(\mathcal{A}) = q'\sqrt{\frac{q}{2^{2n}}}$ as long as the q online queries are done classically. This was already done essentially in this context in [37], where they showed how to reprogram a quantum oracle having offline queries to E and E^{-1} using quantum one-way to hiding theorems.

2.5 Quantum Computing

We will discuss some basic quantum algorithms and observations that we will use in this paper. Performing a quantum query to a function f means applying the unitary $O_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$. If f is efficiently computable classically then O_f is efficiently computable quantumly. If f is a permutation, we can also use $IN_f : |x\rangle \to |f(x)\rangle$, which is efficiently computable if both f and f^{-1} are classically efficiently computable.[6]

We remark that in the quantum setting, different attack models are possible. The Q1 setting allows the attacker to use a quantum computer but it can make

[6] This is done by computing $|x\rangle|0\rangle \xrightarrow{O_f} |x\rangle|f(x)\rangle \xrightarrow{\text{swap}} |f(x)\rangle|x\rangle \xrightarrow{O_{f^{-1}}} |f(x)\rangle|0\rangle$.

only classical queries to the black-box keyed oracles. In the Q2 setting, the attacker is able to make superposition queries to the black-box keyed primitives. Various attacks in both settings have appeared over time. In particular, the Q2 setting allows to attack many symmetric cryptographic algorithms [21,39,41,42]. As the Q2 setting is the strongest of the two, and also represents security in other weaker scenarios, we will aim for resistance of our construction in this setting.

Simon's Algorithm. A function $g : \{0,1\}^n \to \{0,1\}^m$ is said to have a period s when $g(x) = g(y)$ if and only if $x = y$ or $x = y \oplus s$. If g is efficiently computable, then Simon's algorithm [59] is able to recover s in time $O(n^3)$. A relaxed version of Simon's Algorithm can be used to detect the presence of a period without recovering it [34, Section 4].

It is also possible to only evaluate g on a subspace as long as the subspace admits s as a period: i.e., if x is the subspace, $x \oplus s$ is also in the subspace.

Grover's Search. Given an efficiently computable function $g : \{0,1\}^n \to \{0,1\}$, Grover's search algorithm [30] finds an element x (if it exists) such that $g(x) = 1$ in time $O(2^{n/2})$.

BHT Algorithm. Given a random function $g : \{0,1\}^n \to \{0,1\}^n$, the BHT algorithm [10] finds a collision (i.e., $x \neq y$ such that $g(x) = g(y)$) with $O(2^{n/3})$ quantum queries to g. If g is efficiently computable then the quantum running time is also $O(2^{n/3})$, given access to quantum RAM operations.

It is possible to modify the procedure to get the uniform superposition of collisions instead of a random one.

3 Encrypt-Mix-Encrypt in Quantum Setting

Our aim is to find a $2n$-bit-to-$2n$-bit encryption mode using an n-bit block cipher. We will start from the general Encrypt-Mix-Encrypt construction, which we discuss in Section 3.1. Then, we will describe a new superposition attack on a large class of Encrypt-Mix-Encrypt-style constructions in Section 3.2.

3.1 Generic Construction

Let $E : \{0,1\}^k \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be a block cipher. In the Encrypt-Mix-Encrypt paradigm, the plaintext is first passed through a encryption layer, then an invertible mixing layer with possibly non-linear components, and then another encryption layer. The encryption layers can be weak and simple, such as an ECB layer (with different keys). In this case, the Encrypt-Mix-Encrypt construction operates as follows:

$$(L, R) \mapsto \Big(E_3\big(M(E_1(L), E_2(R))_\ell\big), E_4\big(M(E_1(L), E_2(R))_r\big)\Big),$$

where E_i (for $i = 1, \ldots, 4$) is shorthand notation for $E(K_i, \cdot)$ for some secret key K_i, M is a $2n$-to-$2n$-bit mixing layer with $M(\cdot,\cdot)_\ell$ and $M(\cdot,\cdot)_r$ indicating the left

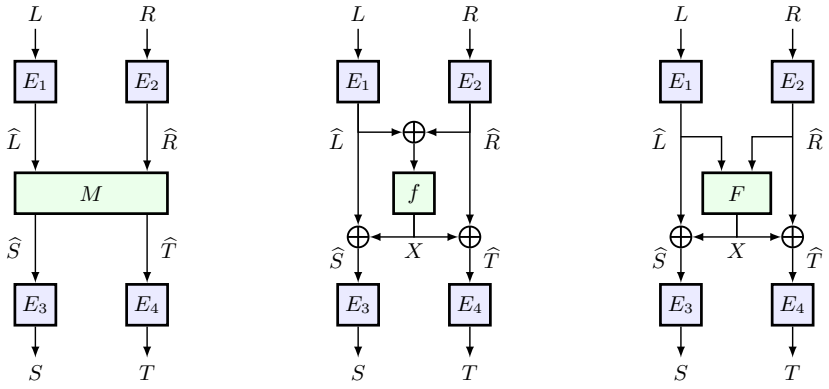


Fig. 2: Variants of the Encrypt-Mix-Encrypt (EME) construction. Figure 2a depicts the construction with generic invertible mixing layer $M : \{0,1\}^{2n} \longrightarrow \{0,1\}^{2n}$. An instantiation of this mixing layer using a non-compressing function $f : \{0,1\}^{n} \longrightarrow \{0,1\}^{n}$ is depicted in Figure 2b and using a compressing function $F : \{0,1\}^{2n} \longrightarrow \{0,1\}^{n}$ in Figure 2c.

and right halves of its output, respectively. This generic Encrypt-Mix-Encrypt construction is depicted in Figure 2a.

The next step is to select a proper invertible mixing function, preferably based on an n-to-n-bit function f (which could, subsequently, be instantiated using a block cipher with a secret key). An example mixing choice of this type would be the Lai-Massey construction [43], as depicted in Figure 2b. In detail, it instantiates the mixing function as

$$M(x, y) := (x \oplus f(x \oplus y), y \oplus f(x \oplus y)). \tag{2}$$

One can consider this construction to be a variant of EME [31].

3.2 Superposition Attack on Wide Class of Variants

Unfortunately, the construction of Figure 2b, i.e., with the mixing of (2), turns out to be insecure in the quantum setting, even if E is a qPRP and f a qPRP or qPRF. Even stronger, we demonstrate that any invertible mixing making a single call to n-to-n-bit function f is insecure in the quantum setting. We demonstrate the result for Figure 2b in Section 3.2.1, and subsequently explain how the result generalizes for arbitrary mixing function in Section 3.2.3.

3.2.1 Attack on Construction of Figure 2b. We describe a general attack that recovers one of the keys of the outer permutations in around $\tilde{O}(2^{n/2})$ time.

Theorem 1. *Let* $K_1, \ldots, K_4 \in \{0,1\}^n$ *be four keys, and denote* $\boldsymbol{K} = (K_1, \ldots, K_4)$ *for brevity. There exists a quantum key recovery adversary* $\mathcal{A}$ *against* $\mathrm{EME}[E, f]_{\boldsymbol{K}}$ *of Figure 2b with the mixing layer of (2) for good function* f *that makes* $\tilde{O}(2^{n/3})$ *queries, operates in* $\tilde{O}(2^{n/2})$ *time and* $\tilde{O}(2^{n/3})$ *memory, and succeeds with probability at least* $1/e$.

The proof is given in Section 3.2.2. Note that the functions f that make the attack fail are affine (i.e., $f(x) \oplus f(x \oplus t) = f(y) \oplus f(y \oplus t)$) on a significant part of inputs. In this case, the construction can instead be attacked by searching collisions on the left side of the output and colliding keys K_1 and K_2 that match the linearity.

3.2.2 Proof of Theorem 1. The idea of the attack is to apply the BHT algorithm to obtain a superposition of pairs of states, each for fixed left inputs L_0 and L_1, that collide on their left half S. This phase runs in time $O(2^{n/3})$. Then, Grover's algorithm is evaluated to obtain the key K_2, which succeeds in time $O(2^{n/2})$. Within this key search, Simon's algorithm is employed to verify correct key guesses. Due to symmetry of the $\mathrm{EME}[E, f]$ mode, the same attack can be applied to recover the keys K_1, K_3, or K_4. The attack is unique in its kind as it combines BHT, Grover, and Simon, where particularly BHT is used to restraint the analyzed function to interesting outputs that generate a partial collision, and Grover and Simon are combined to obtain a more targeted key recovery on top of the earlier collision search.

Description of the Attack. Write $E_i = E_{K_i}$ as shorthand notation. Define the function $S(L, R)$ that on input of (L, R) outputs the left half of $\mathrm{EME}[E, f]_{\boldsymbol{K}}$:

$$S(L, R) = E_3\left(E_1(L) \oplus f(E_1(L) \oplus E_2(R))\right).$$

Next, we fix two distinct values L_0 and L_1, and consider the uniform superposition of claws between $F_0 : R \mapsto S(L_0, R)$ and $F_1 : R \mapsto S(L_1, R)$. The claws are interesting as, actually,

$$\begin{aligned} &S(L_0, R_0) = S(L_1, R_1) \iff \\ &\qquad f\big(E_1(L_0) \oplus E_2(R_0)\big) \oplus f\big(E_1(L_1) \oplus E_2(R_1)\big) = E_1(L_0) \oplus E_1(L_1). \end{aligned}$$

We obtain a uniform superposition

$$\frac{1}{\sqrt{|\{(R_0, R_1) \mid S(L_0, R_0) = S(L_1, R_1)\}|}} \sum_{(R_0, R_1) \in \mathcal{X}} |R_0, R_1\rangle$$

of elements from

$$\mathcal{X} := \left\{(R_0, R_1) \mid f\big(E_1(L_0) \oplus E_2(R_0)\big) \oplus f\big(E_1(L_1) \oplus E_2(R_1)\big) = E_1(L_0) \oplus E_1(L_1)\right\}. \tag{3}$$

This superposition of claws can be obtained with the BHT algorithm in time $O(2^{n/3})$. The algorithm will be ran $O(n)$ time as we will use multiple claws for confirmation later on, and the below next steps are performed for each result.

We add an extra qubit $|b\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ to the state and apply the controlled exchange $(b, R_0, R_1) \mapsto (b, R_b, R_{1-b})$. As a result, the state becomes the uniform superposition of elements from

$$\mathcal{Y} := \big\{(b, R_0, R_1) \mid f\big(E_1(L_b) \oplus E_2(R_0)\big) \oplus f\big(E_1(L_{1-b}) \oplus E_2(R_1)\big) = E_1(L_0) \oplus E_1(L_1)\big\}\,. \quad (4)$$

We will call the resulting state $|\Phi\rangle$.

The next step is Grover's algorithm to guess key $K^\star$ of E_2. *Assuming* that we guess $K^\star = K_2$ correctly, we can apply $IN_{E_{K^\star}} : |x\rangle \to E_{K^\star}(|x\rangle)$ (since we then efficiently compute E_{K^*} and $E_{K^*}^{-1}$) on the two rightmost registers of $|\Phi\rangle$ and get the superposition

$$\frac{1}{\sqrt{2|\{(R_0, R_1) \mid S(L_0, R_0) = S(L_1, R_1)\}|}} \sum_{(b,R_0,R_1)\in\mathcal{Z}} |b, R_0, R_1\rangle\,,$$

where

$$\mathcal{Z} := \big\{(b, R_0, R_1) \mid f\big(E_1(L_b) \oplus R_0\big) \oplus f\big(E_1(L_{1-b}) \oplus R_1\big) = E_1(L_0) \oplus E_1(L_1)\big\}\,. \quad (5)$$

In this case, the set $\mathcal{Z}$ admits the period $\big(1, E_1(L_0) \oplus E_1(L_1), E_1(L_0) \oplus E_1(L_1)\big)$. In other words, if $(b, R_0, R_1) \in \mathcal{Z}$ then also $\big(b\oplus 1, R_0 \oplus E_1(L_0) \oplus E_1(L_1), R_1 \oplus E_1(L_0) \oplus E_1(L_1)\big) \in \mathcal{Z}$. We thus apply Simon's algorithm on $(b, R_0, R_1) \mapsto R_0 \oplus R_1$, to recover the existence of this period and uncompute the last steps to recover the states $|\Phi\rangle$ if this is the case. We note that the last step may fail for weak choices of f, e.g., if f maps all values to 0. In this case, however, other attacks will be possible.

The attack is described in more detail in Algorithm 1.

Analysis of the Attack. The attack combines the BHT algorithm $O(n)$ times, followed by a combination of Grover's algorithm and Simon's algorithm. The latter part of the attack. The success probability of this second phase of the attack is estimated in Proposition 1 below. In fact, this proposition is slightly more general: it applies to our attack with $g = 0$, $\mathcal{Z} = \big\{(b, R_0, R_1) \mid S(L_0, R_0) = S(L_1, R_1)\big\}$, $f_i' : (b, R_0, R_1) \mapsto (b, E_{2,i}(R_b), E_{2,i}(R_{1-b}))$, $f_i : (b, R, R') \mapsto R' \oplus R$, and $s = (1, E_1(L_0) \oplus E_1(L_1), E_1(L_0) \oplus E_1(L_1))$ and $i_0 = K_2$ where $E_{2,i}$ is E_2 with the key i.

Proposition 1. *Suppose that $m = O(n)$, let $\{f_i : \{0,1\}^n \to \{0,1\}^l\}$ be a family of public functions, and $\{f_i' : \{0,1\}^n \to \{0,1\}^n\}$ family of public permutations. Let $g : A \subseteq \{0,1\}^n \to \{0,1\}^l$ be a function on which we only get some databases $|\phi_g\rangle$. Assume that there is a unique i_0 such that $f_{i_0} \oplus g \circ f_{i_0}'$ has a period s and*

$$\max_{i,t\notin\{0,1\}^m\times\{0\}\cup\{i_0,s\}} \Pr_{x\in f_i'^{-1}(A)} [(f_i \oplus \tilde{g} \circ f_i')(x \oplus t) = (f_i \oplus g \circ f_i')(x)] \le \frac{1}{2}.$$

Algorithm 1 Superposition attack on $\mathrm{EME}[E,f]_{\boldsymbol{K}}$ with the mixing layer of (2)

Input: superposition oracle access to $\mathrm{EME}[E,f]_{\boldsymbol{K}}$
Output: K_2
Select two distinct values $L_0, L_1 \in \{0,1\}^n$
Repeat $O(n)$ **times** (for confirmation)
  Apply BHT algorithm to find claws $F_b : R \mapsto S(L_b, R)$ ▷ $O(2^{n/3})$ time.
    ▷ We obtain uniform superposition $\sum_{(R_0,R_1)\in\mathcal{X}} |R_0, R_1\rangle$ with $\mathcal{X}$ of (3).
EndRepeat
Grover search on K_2 with $O(2^{n/2})$ turns using the following oracle:
  ForEach superposition $\sum_{(R_0,R_1)\in\mathcal{X}} |R_0, R_1\rangle$ of clause 2
    Prepend external qubit $|b\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$
    Apply $(b, R_0, R_1) \mapsto (b, E_{K_2}(R_b), E_{K_2}(R_{1-b}))$
  EndFor
    ▷ If we guessed right, we obtain uniform superpositions on $\mathcal{Z}$ of (5).
    ▷ This set admits the period $\big(1, E_1(L_0) \oplus E_1(L_1), E_1(L_0) \oplus E_1(L_1)\big)$.
  Apply Simon's algorithm on the resulting superpositions
    with function $(b, R_0, R_1) \mapsto R_0 \oplus R_1$
    ▷ Simon's algorithm returns 1 if and only if K_2 is guessed correctly
  Uncompute to retrieve superpositions $\sum_{(R_0,R_1)\in\mathcal{X}} |R_0, R_1\rangle$
EndGrover
Return K_2

Using $O(n)$ databases $|\phi_g\rangle = \sum_{x\in A} \frac{1}{\sqrt{|A|}} |x\rangle |g(x)\rangle$, we can recover i_0 with probability $\Theta(1)$. The running time is $O(n^3 2^{m/2})$.

This result can be obtained as a modification of the offline Simon algorithm [8], which is explained in more detail in Supplementary Material A.

3.2.3 Extension to Arbitrary Mixing Based on Non-Compressing f.

While the attack of Section 3.2.2 is described for the specific mixing of (2), it can readily be adapted to other non-compressing mixing layers similar to (2). Consider a general mixing layer of the following form:

$$M(x,y) := \Pi_2(f(\Pi_1(x,y)), x, y), \tag{6}$$

for some linear maps Π_1 and Π_2. By linearity, we can write $\Pi_1(x,y) = \Pi_{1,L}(x) \oplus \Pi_{1,R}(y)$ and $\Pi_2(f,x,y) = f \oplus \Pi_{2,L}(x) \oplus \Pi_{2,R}(y)$ (even if it means rewriting the function f). The collision equation $S(L_0,R_0) = S(L_1,R_1)$ is then equivalent to

$$\begin{aligned} f(\Pi_{1,L} \circ E_1(L_0) \oplus \Pi_{1,R} \circ E_2(R_0)) \oplus f(\Pi_{1,L} \circ E_1(L_1) \oplus \Pi_{1,R} \circ E_2(R_1)) = \\ \Pi_{2,L} \circ E_1(L_0) \oplus \Pi_{2,L} \circ E_1(L_1) \oplus \Pi_{2,R} \circ E_2(R_0) \oplus \Pi_{2,R} \circ E_2(R_1). \end{aligned}$$

With a good key guess and the controlled exchange, the equation becomes

$$f(\Pi_{1,L} \circ E_1(L_b) \oplus \Pi_{1,R}(R_0)) \oplus f(\Pi_{1,L} \circ E_1(L_{1-b}) \oplus \Pi_{1,R}(R_1)) =$$

$$\Pi_{2,L} \circ E_1(L_0) \oplus \Pi_{2,L} \circ E_1(L_1) \oplus \Pi_{2,R}(R_0) \oplus \Pi_{2,R}(R_1).$$

Now, if $\Pi_{1,R}$ is not reversible, there exists $t \neq 0$ such that $\Pi_{1,R}(t) = 0$, and the set of collisions admits the period $s = (0, t, t)$. On the other hand, if $\Pi_{1,R}$ is reversible, then the set of collisions admits the period $s = (1, t, t)$ for $t = \Pi_{1,R}^{-1} \circ \Pi_{1,L}(E_1(L_0) \oplus E_1(L_1))$. In either case, the attack works the same way but the recovered period will be different.

4 QuEME

The EME construction appears to be like a good starting point for our doubler, however, the attack of Section 3.2 shows that the mixing layer must be chosen with care. In fact, the attack of Section 3.2 excludes all mixing functions based on a single non-compressing primitive f. As our aim is to build our scheme based on a block cipher, the only reasonable alternative is to view the block cipher as a compressing function $E : \{0,1\}^{2n} \longrightarrow \{0,1\}^n$, where one block of n bits goes into the data path and one block of n bits into the key path, and build the mixing layer on top of that.

Unfortunately, also such instantiation must be made with care. Suppose, for example, we would take the arguably most logical choice, namely the specification of Figure 2c, with F replaced by E in such a way that $\widehat{L}$ goes into the key path and $\widehat{R}$ into the data path of E. In this case, one can easily mount a distinguishing attack: An attacker can keep L constant and vary R. This leads to constant $\widehat{L}$, and thus differing X for each query. This also implies that $\widehat{S}$ and thus S differ for each query. On the other hand, for an ideal primitive collisions in S are expected in $2^{n/2}$ evaluations. A logical solution to this approach is to have the key path to E depending on *both* $\widehat{L}$ and $\widehat{R}$. Taking bijectivity of the mixing layer into account, this leads to the mixing layer that we adopted for QuEME:

$$M(x, y) := (E(x \oplus y, x), y \oplus E(x \oplus y, x)).$$

This will be the core idea of QuEME. However, we will describe it in a more general fashion where (i) the block cipher in the mixing layer *may be* different from the block cipher used in the outer layer, and (ii) the block cipher in the outer layer may have a key size different from the data size. In addition, QuEME is defined for four keys. Looking ahead, in Section 9 we propose an instantiation that also deals with how to obtain variation in the block ciphers and in the keys.

In detail, let $E : \{0,1\}^k \times \{0,1\}^n \longrightarrow \{0,1\}^n$ and $E' : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be two block ciphers and let $K_1, \ldots, K_4 \in \{0,1\}^k$ be four keys. Denote $\boldsymbol{K} = (K_1, \ldots, K_4)$. We define $\mathrm{QuEME}^{E,E'} : \{0,1\}^{4k} \times \{0,1\}^{2n} \longrightarrow \{0,1\}^{2n}$ as

$$\mathrm{QuEME}^{E,E'}_{\boldsymbol{K}}(L, R) := (S, T), \tag{7}$$

where

$$\widehat{L} = E(K_1, L), \quad \widehat{R} = E(K_2, R),$$

$$
\begin{aligned}
&X = \widehat{L} \oplus \widehat{R},\\
&\widehat{S} = E'(X, \widehat{L}), \quad \widehat{T} = X \oplus \widehat{S},\\
&S = E(K_3, \widehat{S}), \quad T = E(K_4, \widehat{T}).
\end{aligned}
$$

We simply write QuEME^E in case $k = n$ and $E = E'$. For this case, the scheme is depicted in Figure 1b.

5 Generic Attack in 2^n Queries

We will first describe a generic attack against QuEME$^{E,E'}$ that operates in 2^{n+4} queries. The attack de facto demonstrates that we cannot prove security of QuEME beyond 2^n.

Proposition 2. *Let $E : \{0,1\}^k \times \{0,1\}^n \longrightarrow \{0,1\}^n$ and $E' : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be two block ciphers. There exists a classical PRP adversary $\mathcal{A}$ against* QuEME$^{E,E'}$ *making 2^{n+4} queries such that*

$$\mathbf{Adv}^{\mathrm{prp}}_{\mathrm{QuEME}^{E,E'}}(\mathcal{A}) \geq \Omega(1).$$

The proof is given in Supplementary Material B.

6 Classical $2n/3$-Bit Security of QuEME

We will first give a classical security proof of QuEME$^{E,E'}$ up to $2^{2n/3}$ queries. The main purpose of this result is to demonstrate how the security of QuEME$^{E,E'}$ reduces to the SPRP security of E and to the random access SPRP security of E' as freshly defined in Section 2.4. In Section 7, we will derive security up to 2^n queries by combining this technique and a mirror theory result.

Theorem 2. *Let $E : \{0,1\}^k \times \{0,1\}^n \longrightarrow \{0,1\}^n$ and $E' : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be two independent block ciphers. For any classical SPRP adversary $\mathcal{A}$ against* QuEME$^{E,E'}$*, making q queries and operating in time t, we have*

$$\mathbf{Adv}^{\mathrm{sprp}}_{\mathrm{QuEME}^{E,E'}}(\mathcal{A}) \leq \frac{1.5q^3}{2^{2n}} + \frac{q^2}{2^{2n}} + 4 \cdot \mathbf{Adv}^{\mathrm{sprp}}_{E}(\mathcal{A}') + \mathbf{Adv}^{\text{ra-sprp}}_{E'}(\mathcal{A}''),$$

for some adversaries $\mathcal{A}', \mathcal{A}''$ making q queries and operating in time $t', t'' \approx t$.

If we restrict our focus to PRP security, where $\mathcal{A}$ cannot make queries to the inverse construction, the same result applies with also only PRP and random access PRP security on the right hand side. The proof of Theorem 2 is given in Section 6.1.

6.1 Proof of Theorem 2

Let $\boldsymbol{K} = (K_1, \ldots, K_4) \in \{0,1\}^{4k}$, and $\Pi \xleftarrow{\$} \mathrm{perm}(2n)$. We consider an adversary $\mathcal{A}$ that aims to distinguish $(\mathrm{QuEME}^{E,E'}_{\boldsymbol{K}})^{\pm}$ from $\Pi^{\pm}$:

$$\mathbf{Adv}_{(\mathrm{QuEME}^{E,E'}_{\boldsymbol{K}})^{\pm};\Pi^{\pm}}(\mathcal{A}). \tag{8}$$

We assume that $\mathcal{A}$ never makes redundant queries, which can either be repetitions of earlier queries or relaying an encryption output to the decryption oracle or vice versa.

6.1.1 Reduction to Ideal Primitives. As a first step, we replace the outer block cipher evaluations $E_{K_1}, \ldots, E_{K_4}$ by random permutations $\pi_1, \ldots, \pi_4 \xleftarrow{\$} \mathrm{perm}(n)$. This is a plain SPRP step and comes at the cost of $4 \cdot \mathbf{Adv}^{\mathrm{sprp}}_{E}(\mathcal{A}')$ for some adversary $\mathcal{A}'$ with the same query complexity and comparable time complexity as $\mathcal{A}$. Denoting $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_4)$ and the resulting construction as $\mathrm{QuEME}^{\boldsymbol{\pi},E'}$ for brevity, we obtain

$$(8) \le \mathbf{Adv}_{(\mathrm{QuEME}^{\boldsymbol{\pi},E'})^{\pm};\Pi^{\pm}}(\mathcal{A}) + 4 \cdot \mathbf{Adv}^{\mathrm{sprp}}_{E}(\mathcal{A}'). \tag{9}$$

Next, the interface that $\mathcal{A}$ has towards the inner block cipher exactly matches the ra-SPRP security of E': it can evaluate E' offline, and it can trigger online evaluations without seeing them. More formally, let $\tilde{\pi} \xleftarrow{\$} \mathrm{perm}(n,n)$. Given adversary $\mathcal{A}$ that aims to distinguish $(\mathrm{QuEME}^{\boldsymbol{\pi},E'})^{\pm}$ from $(\mathrm{QuEME}^{\boldsymbol{\pi},\tilde{\pi}})^{\pm}$, we can define the adversary $\mathcal{A}''$ against the ra-SPRP security of E' as follows. Adversary $\mathcal{A}'$ collects all construction queries of $\mathcal{A}$, for each construction query (L,R) it evaluates its oracle and returns the responding (K,X,Y), and similarly for each inverse construction query (S,T). At the end, $\mathcal{A}''$ copies the output bit of $\mathcal{A}$. Denote by $\mathcal{E}$ the event that the oracle is queries for a forward and inverse evaluation such that $(K,X,Y) = (K',X',Y')$. Provided that $\mathcal{E}$ does not happen, the outputs of $\mathcal{A}''$ follow the distributional constraints of the oracle of $\mathcal{A}$. Event $\mathcal{E}$ happens with probability at most $\binom{q}{2}/2^{2n}$. Thus,

$$\mathbf{Adv}_{(\mathrm{QuEME}^{\boldsymbol{\pi},E'})^{\pm};(\mathrm{QuEME}^{\boldsymbol{\pi},\tilde{\pi}})^{\pm}}(\mathcal{A}) \le \mathbf{Adv}^{\text{ra-sprp}}_{E'}(\mathcal{A}'') + \frac{\binom{q}{2}}{2^{2n}}.$$

Adversary $\mathcal{A}''$ with the same query complexity and comparable time complexity as $\mathcal{A}$. We thus obtain

$$(8) \le \mathbf{Adv}_{(\mathrm{QuEME}^{\boldsymbol{\pi},\tilde{\pi}})^{\pm};\Pi^{\pm}}(\mathcal{A}) + 4 \cdot \mathbf{Adv}^{\mathrm{sprp}}_{E}(\mathcal{A}') + \mathbf{Adv}^{\text{ra-sprp}}_{E'}(\mathcal{A}'') + \frac{\binom{q}{2}}{2^{2n}}. \tag{10}$$

We perform one more oracle transformation. Let $f^+, f^- \xleftarrow{\$} \mathrm{func}(2n,2n)$ be two random functions. Then, $f^{\pm} = (f^+, f^-)$ behaves identically to $\Pi^{\pm}$, conditioned on the event that an output of one of the two functions never collides with one

of its previous outputs or with a previous query to the other function (here, we use that $\mathcal{A}$ never makes redundant queries). We can thus perform this RP-RF switch at a cost of $\binom{q}{2}/2^{2n}$:

$$(8) \leq \mathbf{Adv}_{(\mathrm{QuEME}^{\boldsymbol{\pi},\tilde{\pi}})^{\pm};f^{\pm}}(\mathcal{A}) + 4 \cdot \mathbf{Adv}^{\mathrm{sprp}}_{E}(\mathcal{A}') + \mathbf{Adv}^{\text{ra-sprp}}_{E'}(\mathcal{A}'') + \frac{q^2}{2^{2n}}. \tag{11}$$

In the remainder, we will focus on the remaining distance between $\mathcal{O} = (\mathrm{QuEME}^{\boldsymbol{\pi},\tilde{\pi}})^{\pm}$ and $\mathcal{P} = f^{\pm}$ and use the H-coefficient technique (Lemma 1) to bound it.

6.1.2 Additional Notation. We will relax the setting and assume that $\mathcal{A}$ has unbounded computational power and we measure its complexity only by the number of oracle calls it makes. All queries are recorded in a transcript $\tau = \{(L^i, R^i, S^i, T^i, \mathrm{d}^i) \mid i \in [1..q]\}$, with $\mathrm{d}^i \in \{\mathrm{fwd}, \mathrm{inv}\}$ indicating the query direction. We partition $[1..q]$ into $\mathcal{I}^*$, containing the query indices where both output blocks are *fresh*, and $\mathcal{I}$, containing the query indices where one of the output blocks collides with an earlier block at the same position.

We expand the transcript with $\tau^* = \{(\widehat{L}^i, \widehat{R}^i, X^i, \widehat{S}^i, \widehat{T}^i) \mid i \in [1..q]\}$. In the real world $\mathcal{O}$, these are the actual values within the evaluation of $\mathrm{QuEME}^{\boldsymbol{\pi},\tilde{\pi}}$. In the ideal world, these are generated by sampler $\mathcal{S}$ as follows.

1. $\mathcal{S}$ initializes four initially empty tables $D_{\widehat{L}}$, $D_{\widehat{R}}$, $D_{\widehat{S}}$, and $D_{\widehat{T}}$;
2. For each encryption query i with $\mathrm{d}^i = \mathrm{fwd}$:
   (a) $\mathcal{S}$ first checks the tables $D_{\widehat{L}}$ and $D_{\widehat{R}}$ to see if $\widehat{L}^i$ or $\widehat{R}^i$ have already been sampled; whichever is not found in the table is freshly sampled from the set of unsampled values (so outside $D_{\widehat{L}}$ or $D_{\widehat{R}}$, respectively) and added to the corresponding table. X^i is defined to be $\widehat{L}^i \oplus \widehat{R}^i$;
   (b) If $i \in \mathcal{I}$, one of $\widehat{S}^i$ and $\widehat{T}^i$ is already sampled. The other one is defined such that $\widehat{S}^i \oplus \widehat{T}^i = X^i$ and added to the corresponding table;
   (c) If $i \in \mathcal{I}^*$, both $\widehat{S}^i$ and $\widehat{T}^i$ are fresh. The value $\widehat{S}^i$ is sampled from outside $D_{\widehat{S}}$, such that $\widehat{T}^i = \widehat{S}^i \oplus X^i$ is also outside $D_{\widehat{T}}$. Both are added to the corresponding tables;
3. For each decryption query i with $\mathrm{d}^i = \mathrm{inv}$:
   (a) $\mathcal{S}$ first checks the tables $D_{\widehat{S}}$ and $D_{\widehat{T}}$ to see if $\widehat{S}^i$ or $\widehat{T}^i$ have already been sampled; whichever is not found in the table is freshly sampled from the set of unsampled values (so outside $D_{\widehat{S}}$ or $D_{\widehat{T}}$, respectively) and added to the corresponding table. X^i is defined to be $\widehat{S}^i \oplus \widehat{T}^i$;
   (b) If $i \in \mathcal{I}$, one of $\widehat{L}^i$ and $\widehat{R}^i$ is already sampled. The other one is defined such that $\widehat{L}^i \oplus \widehat{R}^i = X^i$ and added to the corresponding table;
   (c) If $i \in \mathcal{I}^*$, both $\widehat{L}^i$ and $\widehat{R}^i$ are fresh. The value $\widehat{L}^i$ is sampled from outside $D_{\widehat{L}}$, such that $\widehat{R}^i = \widehat{L}^i \oplus X^i$ is also outside $D_{\widehat{R}}$. Both are added to the corresponding tables.

6.1.3 Analysis of Idealized QuEME. In the remainder, we write $N = 2^n$ for brevity. We define the following bad events:

- bad_0: For some $i, i', i'' \in [1..q]$ with $i > i'$ and $i > i''$, one of the following holds:
  - $\mathrm{d}^i = \text{fwd}$, $S^i = S^{i'}$, and $T^i = T^{i''}$; or
  - $\mathrm{d}^i = \text{inv}$, $L^i = L^{i'}$, and $R^i = R^{i''}$;
- bad_1: For some $i \in \mathcal{I}$ with $\mathrm{d}^i = \text{fwd}$, a previously unsampled $\widehat{S}^i$ or $\widehat{T}^i$ is set to be equal to a previously sampled value in $D_{\widehat{S}}$ or $D_{\widehat{T}}$, respectively;
- bad_2: For some $i \in \mathcal{I}$ with $\mathrm{d}^i = \text{inv}$, a previously unsampled $\widehat{L}^i$ or $\widehat{R}^i$ is set to be equal to a previously sampled value in $D_{\widehat{L}}$ or $D_{\widehat{R}}$, respectively.

We define $\mathsf{bad} = \mathsf{bad}_0 \vee \mathsf{bad}_1 \vee \mathsf{bad}_2$.

Bad Transcripts. We have to analyze the probability that bad occurs in the ideal world. We have

$$\Pr[\mathsf{bad}] \leq \Pr[\mathsf{bad}_0] + \Pr[\mathsf{bad}_1 \mid \neg\mathsf{bad}_0] + \Pr[\mathsf{bad}_2 \mid \neg\mathsf{bad}_0] \ . \tag{12}$$

For bad_0, the two collisions have a joint probability of $1/N^2$ for any choice of the three indices i, i', i''. The number of such choices for the indices is $\binom{q}{2}$ (in case $i' = i''$) plus $\binom{q}{3}$ (in case $i' \neq i''$), summing to at most $q^3/6$. Thus, $\Pr[\mathsf{bad}_0] \leq \frac{q^3}{6N^2}$.

For bad_1, we need one collision with a previous i' for $i \in \mathcal{I}$, and one collision with a previously sampled value at some i''. Again they have a joint probability of $1/N^2$, for any choice of the three indices i, i', i'' with the same constraints. Thus, $\Pr[\mathsf{bad}_1 \mid \neg\mathsf{bad}_0] \leq \frac{q^3}{6N^2}$.

Similarly we can show that $\Pr[\mathsf{bad}_2 \mid \neg\mathsf{bad}_0] \leq \frac{q^3}{6N^2}$.

From (12) we obtain that

$$\Pr[\mathsf{bad}] \leq \frac{q^3}{2N^2}. \tag{13}$$

Good Transcripts. Suppose (τ, τ^*) is a good transcript. Let q_1, q_2, q_3, q_4 be the number of distinct values of L^i, R^i, S^i, T^i, respectively, in τ. Further suppose that in τ^*, there are r distinct values of X^i, with the number of queries they appear in being $t_1, \ldots, t_r$, where $t_1 + \ldots + t_r = q$.

In the real world, for each $j \in [4]$, the probability that π_j is compatible with (τ, τ^*) is $1/(N)_{q_j}$, and the probability that $\widetilde{\pi}$ is compatible with (τ, τ^*) is $1/[(N)_{t_1} \cdot \ldots \cdot (N)_{t_r}]$. Thus,

$$\Pr[\mathcal{D}_{\mathcal{O}} = (\tau, \tau^*)] = \frac{1}{(N)_{q_1} \cdot \ldots \cdot (N)_{q_4} (N)_{t_1} \cdot \ldots \cdot (N)_{t_r}}. \tag{14}$$

In the ideal world, the responses to the adversary queries are sampled first in the online phase, and then the internal transcript is sampled sequentially, query-by-query. Let p_i be the probability of sampling $(\widehat{L}^i, \widehat{R}^i, X^i, \widehat{S}^i, \widehat{T}^i)$ at the i^{th} query, conditioned on the external transcript and the internal transcript of

the first $i-1$ queries. Noting that the probability of realizing τ in the ideal world is $1/N^{2q}$, we have

$$\Pr\left[\mathcal{D}_{\mathcal{P}} = (\tau, \tau^*)\right] = \frac{1}{N^{2q}} \cdot \prod_{i=1}^{q} p_i. \tag{15}$$

Let $(\widehat{L}_1, \ldots, \widehat{L}_{q_1})$ be the unique values in $\{\widehat{L}^i \mid i \in [q]\}$ ordered according to their order of appearance, and similarly for $(\widehat{R}_1, \ldots, \widehat{R}_{q_2})$, $(\widehat{S}_1, \ldots, \widehat{S}_{q_3})$, and $(\widehat{T}_1, \ldots, \widehat{T}_{q_4})$. For each i let j^i, j'^i, l^i, l'^i be such that $\widehat{L}^i = \widehat{L}_{j^i}$, $\widehat{R}^i = \widehat{R}_{j'^i}$, $\widehat{S}^i = \widehat{S}_{l^i}$, and $\widehat{T}^i = \widehat{T}_{l'^i}$.

First consider $i \in \mathcal{I}$ with $\mathrm{d}^i = \mathrm{fwd}$. Then, we have $p_i = 1/(N - j^i + 1)(N - j'^i + 1)$. Similarly, for $i \in \mathcal{I}$ with $\mathrm{d}^i = \mathrm{inv}$, we have $\widehat{T}^i = \widehat{T}_{l'^i}$, we have $p_i = 1/(N - l^i + 1)(N - l'^i + 1)$. Next consider $i \in \mathcal{I}^*$ with $\mathrm{d}^i = \mathrm{fwd}$. Then, we know that in sampling $\widehat{S}^i$, at most $l^i + l'^i - 2$ values had to be avoided. Thus we can say that $p_i \leq 1/(N - j^i + 1)(N - j'^i + 1)(N - l^i - l'^i + 2)$. We observe that

$$\frac{N(N - l^i - l'^i + 2)}{(N - l^i + 1)(N - l'^i + 1)} = 1 - \frac{(l^i - 1)(l'^i - 1)}{(N - l^i + 1)(N - l'^i + 1)} \geq 1 - \frac{2l^i l'^i}{N^2}, \tag{16}$$

where we use the inequalities $l^i, l'^i \leq N(1 - 1/\sqrt{2})$. Thus,

$$p_i \leq \frac{N}{(N - j^i + 1)(N - j'^i + 1)(N - l^i + 1)(N - l'^i + 1)} \cdot \left(1 - \frac{2l^i l'^i}{N^2}\right)^{-1}. \tag{17}$$

Similarly, for $i \in \mathcal{I}^*$ with $\mathrm{d}^i = \mathrm{inv}$ we have the bound

$$p_i \leq \frac{N}{(N - j^i + 1)(N - j'^i + 1)(N - l^i + 1)(N - l'^i + 1)} \cdot \left(1 - \frac{2j^i j'^i}{N^2}\right)^{-1}. \tag{18}$$

Putting the various cases together we have

$$\begin{aligned}
&\Pr\left[\mathcal{D}_{\mathcal{P}} = (\tau, \tau^*)\right] \\
&\quad \leq \frac{1}{N^{2q}} \cdot \prod_{\substack{i \in \mathcal{I} \\ \mathrm{d}^i = \mathrm{fwd}}} \frac{1}{(N - j^i + 1)(N - j'^i + 1)} \cdot \prod_{\substack{i \in \mathcal{I} \\ \mathrm{d}^i = \mathrm{inv}}} \frac{1}{(N - l^i + 1)(N - l'^i + 1)} \\
&\qquad \cdot \prod_{\substack{i \in \mathcal{I}^* \\ \mathrm{d}^i = \mathrm{fwd}}} \left[\frac{N}{(N - j^i + 1)(N - j'^i + 1)(N - l^i + 1)(N - l'^i + 1)} \cdot \left(1 - \frac{2l^i l'^i}{N^2}\right)^{-1}\right] \\
&\qquad \cdot \prod_{\substack{i \in \mathcal{I}^* \\ \mathrm{d}^i = \mathrm{inv}}} \left[\frac{N}{(N - j^i + 1)(N - j'^i + 1)(N - l^i + 1)(N - l'^i + 1)} \cdot \left(1 - \frac{2j^i j'^i}{N^2}\right)^{-1}\right] \\
&\quad = \frac{1}{N^{2q - |\mathcal{I}^*|}} \cdot \prod_{\substack{i \in \mathcal{I}^* \\ \mathrm{d}^i = \mathrm{fwd}}} \left(1 - \frac{2l^i l'^i}{N^2}\right)^{-1} \cdot \prod_{\substack{i \in \mathcal{I}^* \\ \mathrm{d}^i = \mathrm{inv}}} \left(1 - \frac{2j^i j'^i}{N^2}\right)^{-1}
\end{aligned}$$

$$\cdot \left[\prod_{\mathrm{d}^i=\mathrm{fwd}} \frac{1}{N-j^i+1} \cdot \prod_{\substack{i\in\mathcal{I}^* \\ \mathrm{d}^i=\mathrm{inv}}} \frac{1}{N-j^i+1} \right]$$
$$\cdot \left[\prod_{\mathrm{d}^i=\mathrm{fwd}} \frac{1}{N-j'^i+1} \cdot \prod_{\substack{i\in\mathcal{I}^* \\ \mathrm{d}^i=\mathrm{inv}}} \frac{1}{N-j'^i+1} \right]$$
$$\cdot \left[\prod_{\mathrm{d}^i=\mathrm{inv}} \frac{1}{N-l^i+1} \cdot \prod_{\substack{i\in\mathcal{I}^* \\ \mathrm{d}^i=\mathrm{fwd}}} \frac{1}{N-l^i+1} \right]$$
$$\cdot \left[\prod_{\mathrm{d}^i=\mathrm{inv}} \frac{1}{N-l'^i+1} \cdot \prod_{\substack{i\in\mathcal{I}^* \\ \mathrm{d}^i=\mathrm{fwd}}} \frac{1}{N-l'^i+1} \right]$$
$$\leq \frac{1}{N^{q+|\mathcal{I}|}} \cdot \left(1-\frac{q^3}{N^2}\right)^{-1}$$
$$\cdot \prod_{i\in\mathcal{J}_1} \frac{1}{N-j^i+1} \cdot \prod_{i\in\mathcal{J}_1} \frac{1}{N-j'^i+1} \cdot \prod_{i\in\mathcal{J}_2} \frac{1}{N-l^i+1} \cdot \prod_{i\in\mathcal{J}_2} \frac{1}{N-l'^i+1}, \tag{19}$$

where the index sets in (19) are defined as $\mathcal{J}_1 := \{i \in [q] \mid \mathrm{d}^i = \mathrm{fwd}\} \cup \{i \in \mathcal{I}^* \mid \mathrm{d}^i = \mathrm{inv}\}$ and $\mathcal{J}_2 := \{i \in [q] \mid \mathrm{d}^i = \mathrm{inv}\} \cup \{i \in \mathcal{I}^* \mid \mathrm{d}^i = \mathrm{fwd}\}$.

Now, we observe that the set $\{j^i \mid i \in \mathcal{J}_1\}$ contains all elements in $[q_1]$ except the j for which $\widehat{L}_j$ first appeared at an inverse query index where there was an accidental collision at $\widehat{R}$; let the set of query indices where this happened be called $\mathcal{I}_1$. If we upper bound $N-j+1$ by N for each such j, we thus get

$$\prod_{i\in\mathcal{J}_1} \frac{1}{N-j^i+1} \leq \frac{N^{|\mathcal{I}_1|}}{(N)_{q_1}}. \tag{20}$$

Similarly, if we define $\mathcal{I}_2$ as the set of inverse queries with an accidental collision at $\widehat{L}$, $\mathcal{I}_3$ as the set of forward queries with an accidental collision at $\widehat{T}$, and $\mathcal{I}_4$ as the set of forward queries with an accidental collision at $\widehat{S}$, we have

$$\prod_{i\in\mathcal{J}_1} \frac{1}{N-j'^i+1} \leq \frac{N^{|\mathcal{I}_2|}}{(N)_{q_2}}, \prod_{i\in\mathcal{J}_2} \frac{1}{N-l^i+1} \leq \frac{N^{|\mathcal{I}_3|}}{(N)_{q_3}}, \prod_{i\in\mathcal{J}_2} \frac{1}{N-l'^i+1} \leq \frac{N^{|\mathcal{I}_4|}}{(N)_{q_4}}. \tag{21}$$

The final observation we need is that $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3, \mathcal{I}_4$ form a partition of $\mathcal{I}$, since for each query in $\mathcal{I}$ exactly one accidental collision happens in one of the four places. Putting (20) and (21) in (19) gives

$$\Pr\left[\mathcal{D}_{\mathcal{P}} = (\tau, \tau^*)\right] \leq \frac{1}{N^q \cdot (N)_{q_1} \cdot (N)_{q_2} \cdot (N)_{q_3} \cdot (N)_{q_4}} \cdot \left(1-\frac{q^3}{N^2}\right)^{-1}. \tag{22}$$

From (14) and (22), we get

$$\frac{\Pr\left[\mathcal{D}_{\mathcal{O}} = (\tau, \tau^*)\right]}{\Pr\left[\mathcal{D}_{\mathcal{P}} = (\tau, \tau^*)\right]} \geq \frac{N^q}{(N)_{t_1} \cdot \ldots \cdot (N)_{t_r}} \cdot \left(1 - \frac{q^3}{N^2}\right) \geq 1 - \frac{q^3}{N^2}. \tag{23}$$

Conclusion. From (13) and (23), using the H-coefficient technique of Lemma 1, we obtain that the remaining advantage in (11) is upper bounded by

$$\frac{q^3}{N^2} + \frac{q^3}{2N^2} \leq \frac{1.5q^3}{N^2}.$$

This completes the proof, recalling that $N = 2^n$.

7 Classical n-Bit Security of QuEME

The proof of Section 6 showed clearly how the security of QuEME$^{E,E'}$ reduced to the SPRP of E and random access SPRP security of E', but it only guaranteed security up to $2^{2n/3}$ queries. Using a variant of a mirror theory result, we can reach security up to 2^n queries. This mirror theory result is given in Section 7.1. We state our security result on QuEME$^{E,E'}$ in Section 7.2. The proof is comparable to that of Section 6.1, the main difference being in the fact that we lower bound the probability of good transcripts in the ideal world using the mirror theory lower bound, and we henceforth include the proof in Supplementary Material C. In Supplementary Material D we dive deeper into the mirror theory result of Section 7.1 and perform a simulation, which could be of interest to other mirror theory results as well.

7.1 Mirror Theory

The mirror theory of Patarin [49, 51, 54] gives a lower bound on the number of solutions of systems of bi-variate equations. In its most natural form, it considers q n-bit variables $(Y_1, \ldots, Y_q)$ and r bi-variate equations of the form

$$Y_i \oplus Y_j = \delta_{i,j}, \tag{24}$$

where $i \neq j$ for all equations. It states that, provided all $\delta_{i,j} \neq 0$ and the graph corresponding to these equations (where the variables are nodes and the equations are edges) does not have a cycle or a component larger than $\xi_{\max}$, and provided that $n\xi_{\max}^2 + \xi_{\max} \leq 2^{n/2}$, $r \leq 2^{n/2}$, $r \leq 2^n/12\xi_{\max}$, the number of solutions such that all variables are distinct is at least

$$\frac{(2^n)_q}{2^{nr}}. \tag{25}$$

The intuition behind this lower bound is that the numerator in the above expression is the total number of solutions satisfying just the distinctness constraint, and any randomly chosen solution has a probability of around $1/2^{nr}$ of satisfying

all r bi-variate equations. The mirror theory has a long history [16, 52, 55], and despite having seen various disputes in the community for a long time, it has been well-accepted by the community. There have been various instances in literature where the mirror theory has been used to derive security bounds [6, 36, 48, 65]. Dutta et al. [24] recently provided a clean proof of the bound for $\xi_{\max} = 2$, and Cogliati et al. [15] followed it up with a proof for a wider range of $\xi_{\max}$; the above statement is taken from the latter.

The above statement is useful to obtain a lower bound on the number of permutations $P \in \mathrm{perm}(n)$ which satisfy the r conditions of (24) for a certain q of its outputs. There is also a generalization of this result that considers the case where the n-bit variables come from two different permutations. In other words, the variables $(Y_1, \ldots, Y_q)$ are split over two tuples, say $(Y_1, \ldots, Y_{q_1})$ and $(Z_1, \ldots, Z_{q_2})$, and in each of these two tuples there occurs no collision. The mirror theory variant in this setting asserts that the number of solutions is at least

$$\frac{(2^n)_{q_1}(2^n)_{q_2}}{2^{nr}}. \tag{26}$$

The idea of this one is that, again, the numerator is the total number of solutions satisfying just the distinctness constraint. This variant of the mirror theory has been used in earlier works (e.g., to argue security of EWCDM [48]). A proof of this bound for the special case of $\xi_{\max} = 2$ was provided in [24].

However, it appears that if the graph corresponding to the system of bi-variate equations is structured in a certain way, a slightly improved term can be claimed. Suppose the graph can be split into t components $C^{(1)}, \ldots, C^{(t)}$ where $t = q_1 + q_2 - r$. For each $j \in [1..t]$, let $q_1^{(j)}$ (resp. $q_2^{(j)}$) be the number of Y_i's (resp. Z_i's) that appear in $C^{(j)}$. Finally, define the cumulative sums

$$Q_b^{(j)} = \sum_{i=1}^{j-1} q_b^{(i)}$$

for each $b \in \{1, 2\}$ and each $j \in [1..t]$. This system of equations is consistent when there is no path of even length on which the $\delta_{i,j}$ sum to 0, and it does not have any redundancy when the graph has no cycles. In this case, we conjecture that the number of solutions such that all variables $(Y_1, \ldots, Y_{q_1})$ are distinct and all variables $(Z_1, \ldots, Z_{q_2})$ are distinct is at least

$$\frac{1}{2^{nr}} \cdot \prod_{j=1}^{t} \left[\left(2^n - Q_1^{(j)}\right)^{q_1^{(j)}} \left(2^n - Q_2^{(j)}\right)^{q_2^{(j)}} \right]. \tag{27}$$

The intuition behind this extension is the following. As before we randomly choose a valid solution for the Y_i's and the Z_i's, and it satisfies the equations with (roughly) a probability $1/2^{nr}$. However, in this case, the key additional observation is that when choosing the valid solution, instead of ensuring distinctness among all Y_i's and all Z_i's, we just need to ensure that there are no collisions between components. Indeed, since our system of equations is consistent, for any solution that satisfies the equations, within-component distinctness

is automatically ensured. Thus when choosing the $q_1^{(j)}$ Y_i's from the j^{th} component, we just choose them randomly from all the $N - Q_1^{(j)}$ unsampled values, and similarly for the Z_i's. A small-scale simulation of this mirror theory is given in Supplementary Material D.

7.2 Statement of Result

We are now ready to state the n-bit security result on QuEME^E.

Theorem 3. *Let $E : \{0,1\}^k \times \{0,1\}^n \longrightarrow \{0,1\}^n$ and $E' : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be two independent block ciphers. For any classical SPRP adversary $\mathcal{A}$ against* QuEME$^{E,E'}$*, making q queries and operating in time t, we have*

$$\mathbf{Adv}^{\text{sprp}}_{\text{QuEME}^{E,E'}}(\mathcal{A}) \leq \frac{3.5q^2}{2^{2n}} + 4 \cdot \mathbf{Adv}^{\text{sprp}}_{E}(\mathcal{A}') + \mathbf{Adv}^{\text{ra-sprp}}_{E'}(\mathcal{A}''),$$

for some adversaries $\mathcal{A}', \mathcal{A}''$ making q queries and operating in time $t', t'' \approx t$.

Just like for Theorem 2, if we restrict our focus to PRP security, where $\mathcal{A}$ cannot make queries to the inverse construction, the same result applies with also only PRP and random access PRP security on the right hand side.

The proof of Theorem 3 is very similar to the one of Theorem 2. In particular, the reduction to ideal primitives (Section 6.1.1) and additional notation (Section 6.1.2) carries over verbatim. Differences occur at the point of bounding the idealized version in Section 6.1.3. Here, the mirror theory result comes into play. The complete proof is given in Supplementary Material C.

8 Quantum $n/6$-Bit Security of QuEME

Even though the results of Section 6 and Section 7 give guarantees about the security of QuEME$^{E,E'}$ in the classical setting, our ultimate goal was to develop a construction that achieves a certain level of security against quantum attackers. In this section, we will demonstrate that QuEME$^{E,E'}$ achieves $n/6$-bit PRP security in the quantum setting.

The first step is to reduce the quantum security of QuEME$^{E,E'}$ to its classical security, which happens in Theorem 4. In the formulation below, we will consider quantum adversaries and in some intermediate steps, quantum adversaries that are restricted to classical queries. This will be denoted in the subscript of the adversary: $\mathcal{A}_{QQ}$ will correspond to a quantum adversary performing quantum queries and $\mathcal{A}_{QC}$ will correspond to a quantum adversary performing classical queries. We can now state our main theorem of this section.

Theorem 4. *Let $E : \{0,1\}^k \times \{0,1\}^n \longrightarrow \{0,1\}^n$ and $E' : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be two independent block ciphers. For any quantum PRP adversary $\mathcal{A}_{QQ}$ against* QuEME$^{E,E'}$ *performing quantum queries, for any integer parameter $r \geq 1$, we have*

$$\mathbf{Adv}^{\text{q-prp}}_{\text{QuEME}^{E,E'}}(\mathcal{A}_{QQ}) \le \mathbf{Adv}^{\text{prp}}_{\text{QuEME}^{\boldsymbol{\pi},\tilde{\pi}}}(\mathcal{A}'_{QC}) + \mathbf{Adv}^{\text{ra-q-prp}}_{E'}(\mathcal{A}''_{QC}) + 4 \cdot \mathbf{Adv}^{\text{q-prp}}_{E}(\mathcal{B}_{QQ}) + \frac{r^4}{2^{2n}} + O\left(\frac{q^3}{r}\right),$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_4) \xleftarrow{\$} \text{perm}(n)^4$ *and* $\tilde{\pi} \xleftarrow{\$} \text{perm}(n,n)$. *Moreover,* $\mathcal{A}'_{QC}$ *and* $\mathcal{A}''_{QC}$ *are quantum adversaries performing* r^2 *classical queries, and* $\mathcal{B}_{QQ}$ *is a quantum adversary performing as much quantum queries as* $\mathcal{A}_{QQ}$.

The proof of Theorem 4 is given in Section 8.1. The proof consists of first applying the quantum step to qPRP security just like in Section 6.1.1, leaving a randomized scheme $\text{QuEME}^{\boldsymbol{\pi},E'}$ (abusing notation, the keys are irrelevant here), and then reducing the quantum security of that scheme to its classical security using Zhandry's lower bound on small range functions [62]. Entering the PRP security result of Theorem 3 for this scheme into the equation, we obtain an equilibrium for $r = q^{3/5}2^{2n/5}$, which immediately yields the following corollary.

Corollary 1. *For any quantum PRP adversary* $\mathcal{A}_{QQ}$ *against* $\text{QuEME}^{E,E'}$*, making* q *quantum we have,*

$$\mathbf{Adv}^{\text{q-prp}}_{\text{QuEME}^{E,E'}}(\mathcal{A}_{QQ}) \le \mathbf{Adv}^{\text{ra-q-prp}}_{E'}(\mathcal{A}''_{QC}) + 4 \cdot \mathbf{Adv}^{\text{q-prp}}_{E}(\mathcal{B}_{QQ}) + O\left(\frac{q^{12/5}}{2^{2n/5}}\right).$$

where $\mathcal{B}_{QQ}$ *makes* q *quantum queries and* $\mathcal{A}''_{QQ}$ *performs* $q^{6/5}2^{4n/5}$ *classical queries.*

Proof. We use the notations and the statement of the above theorem. Using Theorem 3, we have $\mathbf{Adv}^{\text{prp}}_{\text{QuEME}^{\boldsymbol{\pi},\tilde{\pi}}}(\mathcal{A}'_{QC}) = O(\frac{r^4}{2^{2n}})$, since the algorithms $\mathcal{A}'_{QC}$ performs r^2 classical queries. Plugging this into the statement of Theorem 4 and choosing $r = q^{3/5}2^{2n/5}$, we obtain

$$\mathbf{Adv}_{\text{QuEME}^{E,E'}_{\boldsymbol{K}};\Pi}(\mathcal{A}_{QQ}) \le \mathbf{Adv}^{\text{ra-q-prp}}_{E'}(\mathcal{A}''_{QC}) + 4 \cdot \mathbf{Adv}^{\text{q-prp}}_{E}(\mathcal{B}_{QQ}) + O\left(\frac{q^{12/5}}{2^{2n/5}}\right). \qquad \square$$

The above corollary shows the security of our scheme up to $q = 2^{n/6}$. We discuss its tightness and possible improvements in Section 8.2.

8.1 Proof of Theorem 4

Let $\boldsymbol{K} = (K_1, \ldots, K_4) \in \{0,1\}^{4k}$, and $\Pi \xleftarrow{\$} \text{perm}(2n)$. We consider a quantum adversary $\mathcal{A}_{QQ}$ that aims to distinguish $\text{QuEME}^{E,E'}_{\boldsymbol{K}}$ from Π:

$$\mathbf{Adv}^{\text{q-prp}}_{\text{QuEME}^{E,E'}}(\mathcal{A}_{QQ}) = \mathbf{Adv}_{\text{QuEME}^{E,E'}_{\boldsymbol{K}};\Pi}(\mathcal{A}_{QQ}). \tag{28}$$

By the quantum PRP definition, we can first write

$$\mathbf{Adv}_{\text{QuEME}^{E,E'}_{\boldsymbol{K}};\Pi}(\mathcal{A}_{QQ}) \le \mathbf{Adv}_{\text{QuEME}^{\boldsymbol{\pi},E'};\Pi}(\mathcal{A}_{QQ}) + 4 \cdot \mathbf{Adv}^{\text{q-prp}}_{E}(\mathcal{B}_{QQ}), \tag{29}$$

where $\mathcal{B}_{QQ}$ is a quantum adversary that performs quantum queries that has essentially the same running time and number of queries than $\mathcal{A}$. We now present our main lemma, that reduces the security with quantum queries to the security with classical queries.

Lemma 2. *For any quantum PRP adversary $\mathcal{A}_{QQ}$ performing q quantum queries there exists a quantum PRP adversary $\mathcal{A}'_{QC}$ performing r^2 classical queries such that*

$$\mathbf{Adv}^{\text{q-prp}}_{\text{QuEME}^{\boldsymbol{\pi},E'}}(\mathcal{A}_{QQ}) \leq \mathbf{Adv}^{\text{prp}}_{\text{QuEME}^{\boldsymbol{\pi},E'}}(\mathcal{A}'_{QC}) + O\left(\frac{q^3}{r}\right). \tag{30}$$

The proof of Lemma 2 is given in Supplementary Material E.

Next, we make use of our random access quantum PRP advantage definition. Starting from our quantum adversary $\mathcal{A}'_{QC}$ that performs r^2 classical queries. We can adopt the reductions of Theorem 2 on the replacement of the primitives, but now up to the quantum security of these primitive, and obtain

$$\mathbf{Adv}_{\text{QuEME}^{\boldsymbol{\pi},E'};\text{QuEME}^{\boldsymbol{\pi},\tilde{\pi}}}(\mathcal{A}'_{QC}) \leq \mathbf{Adv}^{\text{ra-q-prp}}_{E'}(\mathcal{A}''_{QC}) + \frac{r^4}{2^{2n}},$$

for $\tilde{\pi} \xleftarrow{\$} \text{perm}(n,n)$, where $\mathcal{A}''_{QC}$ runs in the same quantum time and performs as much classical queries as $\mathcal{A}'_{QC}$. This implies in particular that

$$\mathbf{Adv}^{\text{prp}}_{\text{QuEME}^{\boldsymbol{\pi},E'}}(\mathcal{A}'_{QC}) \leq \mathbf{Adv}^{\text{prp}}_{\text{QuEME}^{\boldsymbol{\pi},\tilde{\pi}}}(\mathcal{A}'_{QC}) + \mathbf{Adv}^{\text{ra-q-prp}}_{E'}(\mathcal{A}''_{QC}) + \frac{r^4}{2^{2n}}. \tag{31}$$

From (29), (30), and (31), we obtain

$$\begin{aligned}\mathbf{Adv}^{\text{q-prp}}_{\text{QuEME}^{E,E'}_{\boldsymbol{K}}}(\mathcal{A}_{QQ}) \leq \mathbf{Adv}^{\text{prp}}_{\text{QuEME}^{\boldsymbol{\pi},\tilde{\pi}}}(\mathcal{A}'_{QC}) + \mathbf{Adv}^{\text{ra-q-prp}}_{E'}(\mathcal{A}''_{QC}) \\ + 4 \cdot \mathbf{Adv}^{\text{q-prp}}_{E}(\mathcal{B}_{QQ}) + \frac{r^4}{2^{2n}} + O\left(\frac{q^3}{r}\right).\end{aligned}$$

which completes the proof of Theorem 4.

8.2 Discussion

What about the $\mathbf{Adv}^{\text{ra-q-prp}}$ term in this setting? In the ideal cipher model, our adversary $\mathcal{A}''_{CQ}$ would make r^2 inner classical queries E' and q offline quantum queries to E' and $(E')^{-1}$. As we discussed in Section 2.4, since $r = q^{3/5}2^{2n/5}$, this implies $\mathbf{Adv}^{\text{ra-q-prp}}_{E'}(\mathcal{A}''_{QC}) \leq \frac{q^{8/5}}{2^{3n/5}}$, which is much smaller than 1 for $q \leq 2^{n/6}$ so this term will not harm the $n/6$ quantum security bits provided our E' block cipher has good post-quantum property for this random access measure.

On a different note, our proof is very generic and relies on the observation that we can relate the quantum security to the classical security for any construction

that start by encrypting the left and right halves of the input. The drawback of this strategy is that it seems to be far from tight. Indeed, when looking at our construction and the classical attack running with $O(2^n)$ queries (Section 5), it is not clear how to use quantum queries to improve this attack. We expect our construction to have much more than $n/6$ bits of quantum security. It is likely that it even achieves n-bit quantum security.

One possible avenue to improve our bound would be to to look at Zhandry's quantum query recording technique [64]. However, in our case, we need to consider random permutations and not random functions, and this is notoriously hard, as some of the proposals for this turned out to be incorrect (see for instance [61]). As this topic becomes more mature, we hope that this tool will be available for proving tight quantum security bounds for our construction.

9 Concrete Instantiation: Double-AES

The ultimate aim of this work is to present a doubler, i.e., to double the block and key size of the underlying block cipher. QuEME doubles the block cipher, but currently needs four different keys $K_1, \ldots, K_4$, and also for its security it is preferable that the outer block cipher behaves independently from the inner one. This is particularly the case if the four keys $K_1, \ldots, K_4$ are related.

In this section, we propose a concrete instantiation based on AES-128 [17]. We show how to derive four 128-bit keys from our main 256-bit key and how to vary the scheme so that this derivation is safe in Section 9.1. We describe our concrete variants depending on how many AES rounds are considered in each block in Section 9.2. The best attacks we found on these constructions are given in Section 9.3. As we will see, these results motivate us to consider including the last MixColumns transformation on each block cipher call. We estimate and compare implementation performances in Section 9.4, among others with Saturnin [11].

We refer to Supplementary Material F for a brief description of AES-128 and a discussion of the best known attacks on it.

9.1 Key Extension and Scheme Variation

In the ongoing instantiation, the key input K is of size 256 bits. We can split it into two, $K = K_1 \| K_2$, to obtain the two 128-bit keys to the block ciphers in the top layer. The keys K_3 and K_4 for the bottom layer will then be derived from K_1 and K_2 in such a way that knowledge of any of the keys does not give any information about any of the other keys; at least two keys are needed to obtain a third one. For this, we propose to take $K_3 = K_1 \oplus K_2$ and $K_4 = K_1 \oplus (K_2 \lll 1)$.

Refer to Supplementary Material F for a brief description of AES-128. As the security proof assumes independence of the four keys, but eventually they are related, it is beneficial to have some variation in the block cipher evaluations.

We resolve this by using different constants for each round in each cipher:

$$rc_{i,j} = \begin{pmatrix} X^i \bmod X^8 + X^4 + X^3 + X + 1 \\ j \\ 0 \\ 0 \end{pmatrix}.$$

For the inner block cipher in the QuEME mode we maintain the original AES definition.

9.2 Concrete Proposals

Double-AES-10. This is QuEME with the key and constant definitions of Section 9.1 with full 10-round AES-128 encryptions for each block cipher.

Double-AES-7. It seems that the 7-round attacks on AES-128 (refer to Table 3) cannot be exploited when using AES-128 in our construction. Even stronger, given the restricted access that the attacker has on the block ciphers within QuEME, it seems that this 7-round version is already quite conservative.

Double-AES-6-MC. We propose a variant where the number of rounds of AES-128 is reduced to 6, but where an additional MixColumns operation is performed at the end of the block ciphers in the top and middle layer. We encourage cryptanalysis of Double-AES-5-MC, i.e., this version but instantiated with 5-round AES-128, for which we think an attack might exist, but we conjecture that Double-AES-6-MC provides a comparable level of security as Double-AES-10.

9.3 Security Claims and Cryptanalysis

We claim that our instantiations achieve the same quantum security as the Saturnin block cipher [11], that was conceived with the objective of proposing resistance against quantum-attackers. In particular, we also claim that there exists no quantum attack in the single-key setting with $T^2/p < 2^{224}$, where T is the time/query complexity, p the success probability. We do not provide security against related-key superposition attacks (as is the case of all known block ciphers).

In addition, we claim that when plugged into a secure mode, any attack that requires a collision on the state would require at least the generic complexity for generating a collision. In other words, no attack significantly better than $T^5 \times M_q = 2^{512}$ exists, where M_q denotes the quantum memory (that includes the classical memory). This is the theoretical limit given by the best generic attacks, as stated in [11].

9.3.1 Cryptanalysis. In the remainder of this section, we present the best attacks we have found on round-reduced versions of Double-AES. In order to reflect that a different number of rounds can be considered per block, we say that an attack is on variant r_1-r_2-r_3 of Double-AES when it covers r_1 rounds for E_1 and E_2, r_2 rounds for E, and r_3 rounds for E_3 and E_4. The two best attacks that we found cover (i) 3 rounds in E, E_3, and E_4, and any number of rounds in E_1 and E_2, which we denote as X-3-3, and (ii) 2 rounds in the E and any number of rounds in the outer block ciphers, which we denote as X-2-X. Both attacks are not considering an added MixColumns layer at E_1, E_2, and E. Also, both attacks are, logically, quantum attacks.

The attacks rely on the core observation that given the key addition operation of AES-128, the data path of the inner block cipher after the first key addition will be equal to the output of E_2. This property is very interesting. For example, if we consider differences and if the right half of the input is fixed, the first SubBytes transformation in the middle block cipher will have no active S-boxes.

Attack on X-3-3 Version. This attack is based on the square attack [25]. From this attack, we know that if we consider four rounds of AES, and we encrypt a set of 2^8 inputs taking all the possible values of a concrete byte (we call this a saturated byte) while fixing the rest of the state, we will obtain 2^8 outputs verifying that the values of all their 16 words are balanced.

We want to exploit a similar property in our attack: if we guess K_1 of E_1, we can generate an input to the middle call E with some saturated bytes through the output of E_1. We have to be careful, however, as this input state will also influence its subkeys. Taking into account the key schedule and difference propagation through the subkeys, we obtain the path in Figure 3 that holds with probability 1. Given an input to E with two saturated bytes, generated from the output of E_1 as shown in the figure, three rounds (without the last MixColumns) later, we obtain a state where 8 bytes take all different values, in green; one balanced byte, in purple; and 6 constant ones.

As we guess K_1 of E_1 to compute the inputs that generated the desired inputs to E, and the input to E_2 is constant, any number of rounds in E_1 and E_2 would allow the attack to work. The output of E from Figure 3 is directly fed as input to E_3. We now guess the subkey bytes from K_3 associated to 32 bits of the antidiagonal for the last subkey, and 8 bits of information from the subkey of the penultimate equivalent subkey, as shown in Figure 4. This allows us to compute, for the 128 leftmost bits of the outputs of Double-AES version X-3-3, a byte after the first MixColumns transformation in E_3, and to check whether the sum of the resulting bytes is 0 or not. This produces a filter that only keeps one guess out of 256 (2^{-8}).

In order to increase this sieving, we choose, instead of one fixed state as input of E_2, 21 different ones, which would provide a sieving with a probability of $2^{-8\times 21}$ to have the 21 bytes associated to a guess balanced, leaving approximately $2^{128+5\times 8} \times 2^{-8\times 21} = 1$ key guess as candidate for the correct value.

The complexity will be $21 \times 2^8 \times 2^{(128+32+8)/2} = 2^{96.5}$ time and $21 \times 2^8 \times 2^{(128)/2} = 2^{76.5}$ data, where 21×2^8 is the number of inputs we will consider per

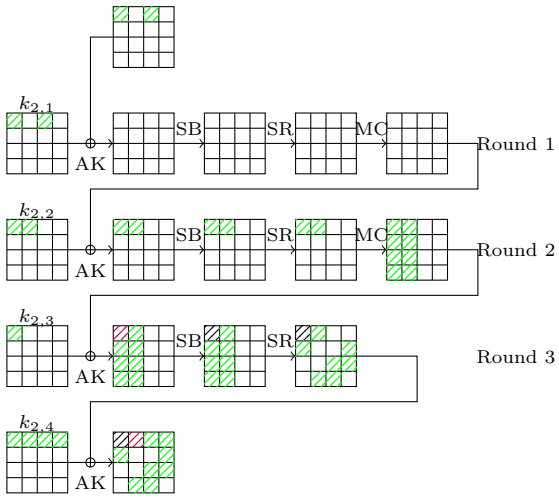


Fig. 3: Square-like property on the middle layer of round-reduced Double-AES. We use green for bytes that take all values (*i.e.* that are saturated, purple for balanced bytes, and black for ignored bytes.

key guess, and $2^{(128+32+8)/2}$ the cost of exhaustive search of the key when done with Grover's algorithm. We expect that these attacks might be extendable to the 4-4-4 configuration, or to 4-3-4 with MixColumns, by having a closer look at the properties generated by the key schedule of the middle layer.

Attack on X-2-X Version. We start with a fixed pair (P_1, P_2) of distinct input blocks to E_1 and perform the encryption through Double-AES version X-2-X of (P_1, R), (P_2, R), for a fixed value R, which is the input to E_2. We consider exclusively the left part of the output, i.e., the output of E_3, and we obtain C_1 and C_2. For each guess of key K_3 from E_3, we will try a decryption through E_3 of C_1 and C_2 and record the difference δ.

In parallel, we guess key K_1 from E_1, and for each guess we will try an encryption through E_1 of P_1 and P_2. This will produce values $\widehat{L}_1$ and $\widehat{L}_2$ that correspond to the values that should enter the middle part E.

Then, we will experience the cancellation of the first round as described above. The second round starts by a subkey addition, and we can get to know the differences on the bytes 0, 4, 8, and 12 (the first line) before the second layer of SubBytes for one additional guess of the byte 13 of $E_2(R)$. Each one of these differences can be associated to $2^{32-4} = 2^{28}$ output differences through the DDT of these four S-boxes. The output differences of the second layer of SubBytes will be determined by δ XOR-ed to the last subkey of the middle layer. In order to compute the difference of this subkey for the first line, and therefore the possible values for finding a match of this first line with the δs, we can perform an additional guess of byte 14 of $E_2(R)$ and the XOR of the

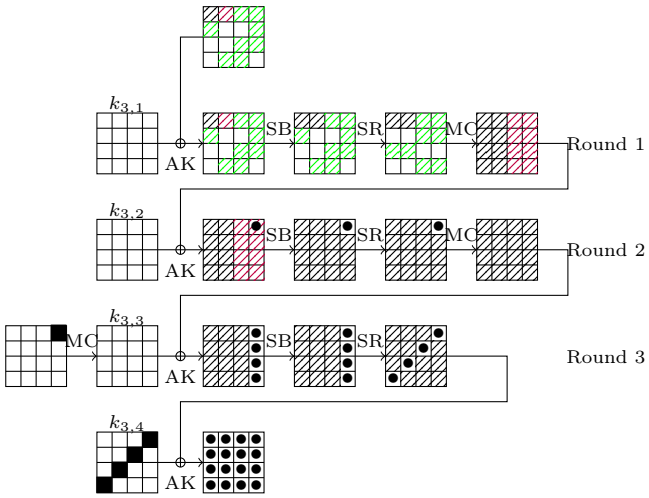


Fig. 4: Recovery on the bottom layer of round-reduced Double-AES. We use green for saturated bytes, purple for balanced bytes, black for ignored bytes, ■ for guessed bytes, and • for deduced and known bytes.

bytes 1, 5, 9, and 13 of $E_2(R)$. We therefore get to compute the possible output differences of E and compare them to the differences δ obtained earlier.

The probability of this sieving is of 2^{-4} because of the DDT. In order to sieve more guesses, we use 70 pairs instead of one, which leaves approximately $2^{128+128+3\times 8} \times 2^{-4\times 70} = 1$ combination.

We can then use an element distinctiveness algorithm to find the correct combination of (K_1, K_3). Thanks to Ambainis algorithm [2], the cost of this attack will be about $70 \times (2^{128} + 2^{128+24})^{2/3} = 2^{107.5}$ time and memory.

9.4 Estimated Implementations Evaluations

Using implementation statistics on the AES round function [58], we can get a fairly reasonable estimation of the implementation costs of our proposed schemes, and we can compare it with Saturnin [11]. It also makes sense to compare our instantiations with Rijndael-256 [17], as it has comparable state and key size. The comparison is given in Table 2. We can observe that in particular Double-AES-6-MC is better than all other variants with respect to cycle count.

10 Conclusion

In this paper, we provide the first proposal of a generic way to double both the key and the state size of a block cipher achieving n-bit security, including both classical and quantum security arguments. As a bonus, we proposed a

Table 2: Estimation of implementation performances.

Cipher	Gates per processed bit	Cycles per block
Rijndael-256-256	283.5	1848
Saturnin	118.5	1678
Double-AES	506.5	1980
Double-AES-7	354.5	1386
Double-AES-6-MC	306.25	1188

new type of superposition attack on the EME construction in Section 3.2, a distinguishing attack matching our security bound, and a method for performing simulations for the mirror theory. We also proposed concrete instantiations of our construction, namely Double-AES, Double-AES-7, and Double-AES-6-MC, along with preliminary cryptanalysis. The instantiations come with a unified security claim regarding classical and quantum attackers. We believe that it is an interesting question to consider security of our instantiations had we reduced the number of rounds even further. Our best attack reaches X-3-3 rounds, i.e., any number of rounds in the first layer, 3 rounds in the middle layer, and 3 rounds in the final layer. An interesting further avenue would be to investigate the power of related-key attacks on AES, noting that the key input to the middle layer varies per evaluation of the scheme.

We believe that our quantum security bound of Section 8 is not tight. It would be interesting to explore the possibilities of using a quantum reduction proof based on a recording oracle, akin to [32]. The main difficulty here is that there is no known way to lazily sample a permutation or to respond to inverse queries using a quantum recording oracle.

References

1. Alagic, G., Bai, C., Katz, J., Majenz, C.: Post-Quantum Security of the Even-Mansour Cipher. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III. Lecture Notes in Computer Science, vol. 13277, pp. 458–487. Springer (2022), https://doi.org/10.1007/978-3-031-07082-2_17
2. Ambainis, A.: Quantum Walk Algorithm for Element Distinctness. SIAM J. Comput. 37(1), 210–239 (2007), https://doi.org/10.1137/S0097539705447311
3. Bao, Z., Guo, J., List, E.: Extended Truncated-differential Distinguishers on Round-reduced AES. IACR Trans. Symmetric Cryptol. 2020(3), 197–261 (2020), https://doi.org/10.13154/tosc.v2020.i3.197-261
4. Bar-On, A., Dunkelman, O., Keller, N., Ronen, E., Shamir, A.: Improved Key Recovery Attacks on Reduced-Round AES with Practical Data and Memory Complexities. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10992, pp. 185–212. Springer (2018), https://doi.org/10.1007/978-3-319-96881-0_7
5. Bellare, M., Krovetz, T., Rogaway, P.: Luby-Rackoff Backwards: Increasing Security by Making Block Ciphers Non-invertible. In: Nyberg, K. (ed.) Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding. Lecture Notes in Computer Science, vol. 1403, pp. 266–280. Springer (1998), https://doi.org/10.1007/BFb0054132
6. Bhattacharjee, A., Bhaumik, R., Nandi, M.: Offset-Based BBB-Secure Tweakable Block-ciphers with Updatable Caches. In: Isobe, T., Sarkar, S. (eds.) Progress in Cryptology - INDOCRYPT 2022 - 23rd International Conference on Cryptology in India, Kolkata, India, December 11-14, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13774, pp. 171–194. Springer (2022), https://doi.org/10.1007/978-3-031-22912-1_8
7. Biryukov, A., Nikolic, I.: Automatic Search for Related-Key Differential Characteristics in Byte-Oriented Block Ciphers: Application to AES, Camellia, Khazad and Others. In: Gilbert, H. (ed.) Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6110, pp. 322–344. Springer (2010), https://doi.org/10.1007/978-3-642-13190-5_17
8. Bonnetain, X., Hosoyamada, A., Naya-Plasencia, M., Sasaki, Y., Schrottenloher, A.: Quantum Attacks Without Superposition Queries: The Offline Simon's Algorithm. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11921, pp. 552–583. Springer (2019), https://doi.org/10.1007/978-3-030-34578-5_20
9. Boura, C., Lallemand, V., Naya-Plasencia, M., Suder, V.: Making the Impossible Possible. J. Cryptol. 31(1), 101–133 (2018), https://doi.org/10.1007/s00145-016-9251-7

10. Brassard, G., Høyer, P., Tapp, A.: Quantum Cryptanalysis of Hash and Claw-Free Functions. In: Lucchesi, C.L., Moura, A.V. (eds.) LATIN '98: Theoretical Informatics, Third Latin American Symposium, Campinas, Brazil, April, 20-24, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1380, pp. 163–169. Springer (1998), https://doi.org/10.1007/BFb0054319
11. Canteaut, A., Duval, S., Leurent, G., Naya-Plasencia, M., Perrin, L., Pornin, T., Schrottenloher, A.: Saturnin: a suite of lightweight symmetric algorithms for post-quantum security. IACR Trans. Symmetric Cryptol. 2020(S1), 160–207 (2020), https://doi.org/10.13154/tosc.v2020.iS1.160-207
12. Chailloux, A., Naya-Plasencia, M., Schrottenloher, A.: An Efficient Quantum Collision Search Algorithm and Implications on Symmetric Cryptography. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10625, pp. 211–240. Springer (2017), https://doi.org/10.1007/978-3-319-70697-9_8
13. Chen, S., Lampe, R., Lee, J., Seurin, Y., Steinberger, J.P.: Minimizing the Two-Round Even-Mansour Cipher. In: Garay, J.A., Gennaro, R. (eds.) Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I. Lecture Notes in Computer Science, vol. 8616, pp. 39–56. Springer (2014), https://doi.org/10.1007/978-3-662-44371-2_3
14. Chen, S., Steinberger, J.P.: Tight Security Bounds for Key-Alternating Ciphers. In: Nguyen, P.Q., Oswald, E. (eds.) Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8441, pp. 327–350. Springer (2014), https://doi.org/10.1007/978-3-642-55220-5_19
15. Cogliati, B., Dutta, A., Nandi, M., Patarin, J., Saha, A.: Proof of Mirror Theory for a Wide Range of $\xi_{\max}$. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 14007, pp. 470–501. Springer (2023), https://doi.org/10.1007/978-3-031-30634-1_16
16. Cogliati, B., Lampe, R., Patarin, J.: The Indistinguishability of the XOR of k Permutations. In: Cid, C., Rechberger, C. (eds.) Fast Software Encryption - 21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers. Lecture Notes in Computer Science, vol. 8540, pp. 285–302. Springer (2014), https://doi.org/10.1007/978-3-662-46706-0_15
17. Daemen, J., Rijmen, V.: Rijndael for AES. In: The Third Advanced Encryption Standard Candidate Conference, April 13-14, 2000, New York, New York, USA. pp. 343–348. National Institute of Standards and Technology, (2000)
18. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography, Springer (2002), https://doi.org/10.1007/978-3-662-04722-4
19. Dai, W., Hoang, V.T., Tessaro, S.: Information-Theoretic Indistinguishability via the Chi-Squared Method. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III. Lecture Notes in Computer

Science, vol. 10403, pp. 497–523. Springer (2017), https://doi.org/10.1007/978-3-319-63697-9_17

20. Derbez, P., Fouque, P., Jean, J.: Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7881, pp. 371–387. Springer (2013), https://doi.org/10.1007/978-3-642-38348-9_23
21. Dong, X., Wang, X.: Quantum key-recovery attack on Feistel structures. Sci. China Inf. Sci. 61(10), 102501:1–102501:7 (2018), https://doi.org/10.1007/s11432-017-9468-y
22. Dunkelman, O., Keller, N., Ronen, E., Shamir, A.: The Retracing Boomerang Attack. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12105, pp. 280–309. Springer (2020), https://doi.org/10.1007/978-3-030-45721-1_11
23. Dunkelman, O., Keller, N., Shamir, A.: Improved Single-Key Attacks on 8-Round AES-192 and AES-256. In: Abe, M. (ed.) Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6477, pp. 158–176. Springer (2010), https://doi.org/10.1007/978-3-642-17373-8_10
24. Dutta, A., Nandi, M., Saha, A.: Proof of Mirror Theory for $\xi_{\max} = 2$. IEEE Trans. Inf. Theory 68(9), 6218–6232 (2022), https://doi.org/10.1109/TIT.2022.3171178
25. Ferguson, N., Kelsey, J., Lucks, S., Schneier, B., Stay, M., Wagner, D.A., Whiting, D.: Improved Cryptanalysis of Rijndael. In: Schneier, B. (ed.) Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings. Lecture Notes in Computer Science, vol. 1978, pp. 213–230. Springer (2000), https://doi.org/10.1007/3-540-44706-7_15
26. Fouque, P., Jean, J., Peyrin, T.: Structural Evaluation of AES and Chosen-Key Distinguisher of 9-Round AES-128. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8042, pp. 183–203. Springer (2013), https://doi.org/10.1007/978-3-642-40041-4_11
27. Gilbert, H.: A Simplified Representation of AES. In: Sarkar, P., Iwata, T. (eds.) Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8873, pp. 200–222. Springer (2014), https://doi.org/10.1007/978-3-662-45611-8_11
28. Grassi, L., Leander, G., Rechberger, C., Tezcan, C., Wiemer, F.: Weak-Key Distinguishers for AES. In: Dunkelman, O., Jr., M.J.J., O'Flynn, C. (eds.) Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers. Lecture Notes in Computer Science, vol. 12804, pp. 141–170. Springer (2020), https://doi.org/10.1007/978-3-030-81652-0_6

29. Grassi, L., Rechberger, C.: Revisiting Gilbert's known-key distinguisher. Des. Codes Cryptogr. 88(7), 1401–1445 (2020), https://doi.org/10.1007/s10623-020-00756-5
30. Grover, L.K.: A Fast Quantum Mechanical Algorithm for Database Search. In: Miller, G.L. (ed.) Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996. pp. 212–219. ACM (1996), https://doi.org/10.1145/237814.237866
31. Halevi, S., Rogaway, P.: A Parallelizable Enciphering Mode. In: Okamoto, T. (ed.) Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings. Lecture Notes in Computer Science, vol. 2964, pp. 292–304. Springer (2004), https://doi.org/10.1007/978-3-540-24660-2_23
32. Hosoyamada, A., Iwata, T.: 4-Round Luby-Rackoff Construction is a qPRP. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11921, pp. 145–174. Springer (2019), https://doi.org/10.1007/978-3-030-34578-5_6
33. Hosoyamada, A., Iwata, T.: Provably Quantum-Secure Tweakable Block Ciphers. IACR Trans. Symmetric Cryptol. 2021(1), 337–377 (2021), https://doi.org/10.46586/tosc.v2021.i1.337-377
34. Ito, G., Hosoyamada, A., Matsumoto, R., Sasaki, Y., Iwata, T.: Quantum Chosen-Ciphertext Attacks against Feistel Ciphers. Cryptology ePrint Archive, Report 2018/1193 (2018), https://eprint.iacr.org/2018/1193
35. Ito, G., Hosoyamada, A., Matsumoto, R., Sasaki, Y., Iwata, T.: Quantum Chosen-Ciphertext Attacks Against Feistel Ciphers. In: Matsui, M. (ed.) Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4-8, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11405, pp. 391–411. Springer (2019), https://doi.org/10.1007/978-3-030-12612-4_20
36. Iwata, T., Mennink, B., Vizár, D.: CENC is Optimally Secure. Cryptology ePrint Archive, Paper 2016/1087 (2016), https://eprint.iacr.org/2016/1087
37. Jaeger, J., Song, F., Tessaro, S.: Quantum Key-Length Extension. In: Nissim, K., Waters, B. (eds.) Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13042, pp. 209–239. Springer (2021), https://doi.org/10.1007/978-3-030-90459-3_8
38. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking Symmetric Cryptosystems Using Quantum Period Finding. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 207–237. Springer (2016), https://doi.org/10.1007/978-3-662-53008-5_8
39. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Quantum Differential and Linear Cryptanalysis. IACR Trans. Symmetric Cryptol. 2016(1), 71–94 (2016), https://doi.org/10.13154/tosc.v2016.i1.71-94
40. Kilian, J., Rogaway, P.: How to Protect DES Against Exhaustive Key Search (an Analysis of DESX). J. Cryptol. 14(1), 17–35 (2001), https://doi.org/10.1007/s001450010015

41. Kuwakado, H., Morii, M.: Quantum distinguisher between the 3-round Feistel cipher and the random permutation. In: IEEE International Symposium on Information Theory, ISIT 2010, June 13-18, 2010, Austin, Texas, USA, Proceedings. pp. 2682–2685. IEEE (2010), https://doi.org/10.1109/ISIT.2010.5513654
42. Kuwakado, H., Morii, M.: Security on the quantum-type Even-Mansour cipher. In: Proceedings of the International Symposium on Information Theory and its Applications, ISITA 2012, Honolulu, HI, USA, October 28-31, 2012. pp. 312–316. IEEE (2012), https://ieeexplore.ieee.org/document/6400943/
43. Lai, X., Massey, J.L.: Hash Function Based on Block Ciphers. In: Rueppel, R.A. (ed.) Advances in Cryptology - EUROCRYPT '92, Workshop on the Theory and Application of of Cryptographic Techniques, Balatonfüred, Hungary, May 24-28, 1992, Proceedings. Lecture Notes in Computer Science, vol. 658, pp. 55–70. Springer (1992), https://doi.org/10.1007/3-540-47555-9_5
44. Leander, G., May, A.: Grover Meets Simon - Quantumly Attacking the FX-construction. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10625, pp. 161–178. Springer (2017), https://doi.org/10.1007/978-3-319-70697-9_6
45. Leurent, G., Pernot, C.: New Representations of the AES Key Schedule. In: Canteaut, A., Standaert, F. (eds.) Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12696, pp. 54–84. Springer (2021), https://doi.org/10.1007/978-3-030-77870-5_3
46. Lu, J., Dunkelman, O., Keller, N., Kim, J.: New Impossible Differential Attacks on AES. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) Progress in Cryptology - INDOCRYPT 2008, 9th International Conference on Cryptology in India, Kharagpur, India, December 14-17, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5365, pp. 279–293. Springer (2008), https://doi.org/10.1007/978-3-540-89754-5_22
47. Lucks, S.: The Sum of PRPs Is a Secure PRF. In: Preneel, B. (ed.) Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding. Lecture Notes in Computer Science, vol. 1807, pp. 470–484. Springer (2000), https://doi.org/10.1007/3-540-45539-6_34
48. Mennink, B., Neves, S.: Encrypted Davies-Meyer and Its Dual: Towards Optimal Security Using Mirror Theory. In: Katz, J., Shacham, H. (eds.) Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III. Lecture Notes in Computer Science, vol. 10403, pp. 556–583. Springer (2017), https://doi.org/10.1007/978-3-319-63697-9_19
49. Patarin, J.: Luby-Rackoff: 7 Rounds Are Enough for $2^{n(1-\epsilon)}$ Security. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2729, pp. 513–529. Springer (2003), https://doi.org/10.1007/978-3-540-45146-4_30
50. Patarin, J.: Security of Random Feistel Schemes with 5 or More Rounds. In: Franklin, M.K. (ed.) Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19,

2004, Proceedings. Lecture Notes in Computer Science, vol. 3152, pp. 106–122. Springer (2004), https://doi.org/10.1007/978-3-540-28628-8_7
51. Patarin, J.: On Linear Systems of Equations with Distinct Variables and Small Block Size. In: Won, D., Kim, S. (eds.) Information Security and Cryptology - ICISC 2005, 8th International Conference, Seoul, Korea, December 1-2, 2005, Revised Selected Papers. Lecture Notes in Computer Science, vol. 3935, pp. 299–321. Springer (2005), https://doi.org/10.1007/11734727_25
52. Patarin, J.: A Proof of Security in $O(2^n)$ for the Xor of Two Random Permutations. In: Safavi-Naini, R. (ed.) Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada, August 10-13, 2008, Proceedings. Lecture Notes in Computer Science, vol. 5155, pp. 232–248. Springer (2008), https://doi.org/10.1007/978-3-540-85093-9_22
53. Patarin, J.: The "Coefficients H" Technique. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers. Lecture Notes in Computer Science, vol. 5381, pp. 328–345. Springer (2008), https://doi.org/10.1007/978-3-642-04159-4_21
54. Patarin, J.: Introduction to Mirror Theory: Analysis of Systems of Linear Equalities and Linear Non Equalities for Cryptography. Cryptology ePrint Archive, Paper 2010/287 (2010), https://eprint.iacr.org/2010/287
55. Patarin, J.: Security of balanced and unbalanced Feistel Schemes with Linear Non Equalities. Cryptology ePrint Archive, Paper 2010/293 (2010), https://eprint.iacr.org/2010/293
56. Rahman, M., Saha, D., Paul, G.: Boomeyong: Embedding Yoyo within Boomerang and its Applications to Key Recovery Attacks on AES and Pholkos. IACR Trans. Symmetric Cryptol. 2021(3), 137–169 (2021), https://doi.org/10.46586/tosc.v2021.i3.137-169
57. Rønjom, S., Bardeh, N.G., Helleseth, T.: Yoyo Tricks with AES. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10624, pp. 217–243. Springer (2017), https://doi.org/10.1007/978-3-319-70694-8_8
58. Schwabe, P., Stoffelen, K.: All the AES You Need on Cortex-M3 and M4. In: Avanzi, R., Heys, H.M. (eds.) Selected Areas in Cryptography - SAC 2016 - 23rd International Conference, St. John's, NL, Canada, August 10-12, 2016, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10532, pp. 180–194. Springer (2016), https://doi.org/10.1007/978-3-319-69453-5_10
59. Simon, D.R.: On the Power of Quantum Computation. SIAM J. Comput. 26(5), 1474–1483 (1997), https://doi.org/10.1137/S0097539796298637
60. Tunstall, M.: Improved "Partial Sums"-based Square Attack on AES. In: Samarati, P., Lou, W., Zhou, J. (eds.) SECRYPT 2012 - Proceedings of the International Conference on Security and Cryptography, Rome, Italy, 24-27 July, 2012, SECRYPT is part of ICETE - The International Joint Conference on e-Business and Telecommunications. pp. 25–34. SciTePress (2012)
61. Unruh, D.: Compressed Permutation Oracles (And the Collision-Resistance of Sponge/SHA3). Cryptology ePrint Archive, Report 2021/062 (2021), https://eprint.iacr.org/2021/062
62. Zhandry, M.: A note on the quantum collision and set equality problems. Quantum Inf. Comput. 15(7&8), 557–567 (2015), https://doi.org/10.26421/QIC15.7-8-2

63. Zhandry, M.: A Note on Quantum-Secure PRPs. Cryptology ePrint Archive, Report 2016/1076 (2016), https://eprint.iacr.org/2016/1076
64. Zhandry, M.: How to Record Quantum Queries, and Applications to Quantum Indifferentiability. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II. Lecture Notes in Computer Science, vol. 11693, pp. 239–268. Springer (2019), https://doi.org/10.1007/978-3-030-26951-7_9
65. Zhang, P., Hu, H., Yuan, Q.: Close to Optimally Secure Variants of GCM. Secur. Commun. Networks 2018, 9715947:1–9715947:12 (2018), https://doi.org/10.1155/2018/9715947

SUPPLEMENTARY MATERIAL

A Offline Simon Algorithm

We start by recalling the Hadamard gate. The Hadamard gate (H) maps $|0\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|1\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad |b\rangle \text{ —— } \boxed{H} \text{ —— } \frac{1}{\sqrt{2}}(|0\rangle + (-1)^b |1\rangle)$$

Using the Hadamard gate, we can discuss the fundamentals of Simon's algorithm [59]. Simon's algorithm is described in Algorithm 2

Simon's algorithm consists in applying Simon's routine of Algorithm 2 $l = O(n)$ times, thus getting $(y_1, \ldots, y_l)$ and solving the following linear system with unknown s:

$$\begin{cases} y_1 \cdot s = 0 \\ \quad \vdots \\ y_l \cdot s = 0 \end{cases}$$

This version of Simon's algorithm requires as a premise that g is a two-to-one function. Luckily, it has also been studied for random functions that admit a period.

Theorem 5 (Kaplan et al. [38, Theorem 2]). *Suppose that $g : \{0,1\}^n \to X$ has a period s, i.e., $g(x \oplus s) = g(x)$ for all $x \in \{0,1\}^n$, and that*

$$\max_{t \notin \{0,s\}} \Pr\left[g(x \oplus t) = g(x)\right] \leq \frac{1}{2}.$$

If we apply Simon's algorithm to g with cn calls to the routine, it returns s with probability at least $1 - 2^n \cdot (3/4)^{cn}$. It runs in cn queries to g and time cn^2.

An important remark on Simon's routine (and on Simon's algorithm by consequence) is that we do not need g if we have access to cn superposition states $|\phi_g\rangle = \sum_{x \in \{0,1\}^n} \frac{1}{2^{n/2}} |x\rangle |g(x)\rangle$. Moreover, we do not need the superposition

Algorithm 2 Description of Simon's routine

Input: superposition oracle access to g
Output: vector y such that $y \cdot s = 0$

1: Initialize state $|0^n\rangle |0^m\rangle$
2: Apply Hadamard gate on all qubits of the first register, obtaining $\frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} |x\rangle \otimes |0^m\rangle$
3: Apply oracle $O_g : |x\rangle |y\rangle \mapsto |x\rangle |y \oplus g(x)\rangle$ to the state, obtaining $\sum_{x\in\{0,1\}^n} \frac{1}{2^{n/2}} |x\rangle |g(x)\rangle$
4: Measure second register and get a value $c = g(x_0)$ for a unknown x_0
▷ By the premise, we get state $\frac{1}{\sqrt{2}} (|x_0\rangle + |x_0 \oplus s\rangle)$.
5: Apply Hadamard gate on all qubits and obtain the state

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{y\in\{0,1\}^n} \left((-1)^{x_0\cdot y} + (-1)^{(x_0\oplus s)\cdot y}\right) |y\rangle$$

▷ This simplifies to

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{y\in\{0,1\}^n} (-1)^{x_0\cdot y} \underbrace{(1 + (-1)^{s\cdot y})}_{0 \text{ if } y\cdot s=1} |y\rangle .$$

6: Measure the state and get a uniformly random y such that $y \cdot s = 0$
7: **Return** y

to include all x in $\{0,1\}^n$; it is possible to restrict g to a subset A as long as this subset admits s as a period, i.e., $x \in A$ if and only if $x \oplus s \in A$, and A does not make an artificial period appear (by restricting on elements such that $g(x \oplus t) = g(x)$ for a certain t). This can be taken to an extreme where $g = 0$ but A has the information of the period.

Corollary 2. *Suppose that $g : A \subseteq \{0,1\}^n \to X$ has a period s, i.e., $x \oplus s \in A$ for all $x \in A$, and that*

$$\max_{t\notin\{0,s\}} \Pr\left[g(x \oplus t) = g(x)\right] \leq \frac{1}{2}. \max_{t\notin\{0,s\}} \Pr_{x\in A}\left[\tilde{g}(x \oplus t) = g(x)\right] \leq \frac{1}{2},$$

where

$$\tilde{g}(x) = \begin{cases} g(x) & \text{if } x \in A, \\ \perp & \text{otherwise.} \end{cases}$$

If we apply Simon's algorithm to cn copies of $|\phi_g\rangle = \sum_{x\in A} \frac{1}{\sqrt{|A|}} |x\rangle |g(x)\rangle$, it returns s with probability at least $1 - 2^n \cdot (3/4)^{cn}$. It runs in time cn^2.

Finally, because the properties of Simon's algorithm did not change because of the input restriction on g, we can apply the ideas of offline Simon's algorithm [8] of Algorithm 3. Here, we define *RANK* to be a circuit that takes $|y_1\rangle \cdots |y_l\rangle |b\rangle$ and flips b if and only if the previous system admits a solution other than 0.

Theorem 6 (Bonnetain et al. [8, Proposition 2]). *Suppose that $m = O(n)$ and let $f : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^l$ be a public function. Let $g : \{0,1\}^n \to \{0,1\}^l$ be a function on which we only get some databases $|\phi_g\rangle$. Assume that there is a unique i_0 such that $f_{i_0} \oplus g$ has a period s and*

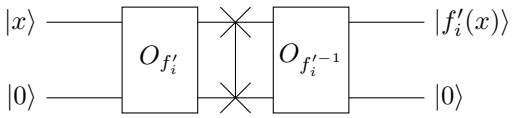
$$\max_{i,t \notin \{0,1\}^m \times \{0\} \cup \{i_0,s\}} \Pr\left[(f_i \oplus g)(x \oplus t) = (f_i \oplus g)(x)\right] \leq \frac{1}{2}.$$

If we apply the offline Simon's algorithm to $O(n)$ databases $|\phi_g\rangle$, it returns i_0 with probability $\Theta(1)$. It runs in time $O(n^3 2^{m/2})$.

Algorithm 3 Description of the Offline-Simon algorithm

Input: superposition oracle access to f and $O(n)$ databases $|\phi_g\rangle$
Output: i_0
Grover search on i with $O(2^{m/2})$ turns using the following oracle:
  Compute $O(n)$ copies of $|\phi_{f_i \oplus g}\rangle = O_{f_i} |\phi_g\rangle$
  Apply Hadamard gate on all qubits of the first registers of $|\phi_{f_i \oplus g}\rangle$, obtaining $O(n)$ states y
    ▷ $y \cdot s = 0$ if $i = i_0$, and random otherwise.
  Apply the *RANK* circuit on the states y
    ▷ A flip occurs if and only if $i = 0$.
  Uncompute the Hadamard gates and O_{f_i} to retrieve databases $|\phi_g\rangle$
EndGrover
Measure and return i

This technique relies on the equality $|\phi_{f_i \oplus g}\rangle = O_{f_i} |\phi_g\rangle$ for preparing and recovering the databases $|\phi_g\rangle$. In our case, instead of $f_{i_0} \oplus g$ being periodic, we look for $f_{i_0} \oplus g \circ f'_{i_0}$ being periodic with f'_i as public permutations. We build the operator $IN_{f'_i} : |x\rangle \mapsto |f'_i(x)\rangle$ using Ancilla qubits and the following circuit:



This allows us to compute $\left|\phi_{f_{i_0} \oplus g \circ f'_{i_0}}\right\rangle = O_{f_i} \circ (IN_{f'^{-1}_i} \otimes I) |\phi_g\rangle$.

This property combined with our observation on input restrictions give the Proposition 1.

B Proof of Proposition 2 (Generic Attack in 2^n Queries)

We will describe an adversary $\mathcal{A}$ that is given access to either $f = \mathrm{QuEME}^{E,E'}_{\boldsymbol{K}}$ (with $\boldsymbol{K} = (K_1, \ldots, K_4)$) or $f = \Pi \xleftarrow{\$} \mathrm{perm}(2n)$. We will write E_i (for $i = 1, \ldots, 4$) as shorthand notation for $E(K_i, \cdot)$. The adversary will only make forward

queries to f, and will be able to distinguish with high probability with which function it communicates. It relies on the core idea that for $f = \Pi$, on any input (L, R), the output $f(L, R) = (S, T)$ is a uniformly random string that has not been output before, whereas if $f = \mathrm{QuEME}^{E,E'}_{\boldsymbol{K}}$, we have

$$E_1(L) \oplus E_2(R) = E_3^{-1}(S) \oplus E_4^{-1}(T). \tag{32}$$

We will use linear algebra techniques in order to detect this structure. Let $N = 2^n$ and $N' = 4N$. Let $L, R, S, T \in \{0,1\}^n$, which we interpret as integers in $[0..N-1]$, and denote by $\boldsymbol{e}_{LRST} \in \mathbb{F}_2^{N'}$ the binary column vector where the i^{th} coordinate of $\boldsymbol{e}_{LRST}$ is equal to 1 if $i = L$, $i = R + N$, $i = S + 2N$ or $i = T + 3N$, and is equal to 0 otherwise. This means each $\boldsymbol{e}_{x_1x_2y_1y_2} \in \mathbb{F}_2^{N'}$ has weight 4, meaning four non-zero coordinates.

Description of the Adversary. The idea of the adversary is the following: perform q queries of the form $\{L^iR^iS^iT^i\}_{i\in[1..q]}$, and let $H = \mathrm{span}\{\boldsymbol{e}_{L^iR^iS^iT^i}\}_{i\in[1..q]}$. For q large enough, but linear in N', we will show that if $\mathcal{A}$ queries $f = \mathrm{QuEME}^{E,E'}_{\boldsymbol{K}}$ construction, we have $\dim(H) \leq N' - 2$ with overwhelming probability, due to (32). On the other hand, if $\mathcal{A}$ queries $f = \Pi$ we have $\dim(H) \geq N' - 1$ since the $\boldsymbol{e}_{L^iR^iS^iT^i}$'s will essentially be random vectors of $\mathbb{F}_2^{N'}$ of weight 4.

More detailed, the adversary operates as follows:

- Perform $q = 4N'$ random different queries (L^i, R^i) for $i \in [1..q]$ and get respective outputs $(S^i, T^i) = f(L^i, R^i)$;
- Let $H = \mathrm{span}\{\boldsymbol{e}_{L^iR^iS^iT^i}\}_{i\in[1..q]}$ and compute $\dim(H)$;
- If $\dim(H) \leq N' - 2$, return "$\mathrm{QuEME}^{E,E'}_{\boldsymbol{K}}$", else return "$\Pi$".

Analysis of the Attack. We will prove in Lemma 3 below that in the real world we always have $\dim(H) \leq N' - 2$ and in Lemma 4 below that in the ideal world we have $\dim(H) = N' - 1$ with overwhelming probability. These two results imply that after $q = 4N' = 2^{n+4}$ queries, $\mathcal{A}$ distinguishes between $\mathrm{QuEME}^{E,E'}_{\boldsymbol{K}}$ and Π with overwhelming probability.

Lemma 3. *If $f = \mathrm{QuEME}^{E,E'}_{\boldsymbol{K}}$, we have $\dim(H) \leq N' - 2$.*

Proof. By construction, each query $(S^i, T^i) = f(L^i, R^i)$ satisfies

$$E_1(L^i) \oplus E_2(R^i) = E_3^{-1}(S^i) \oplus E_4^{-1}(T^i).$$

Consider the following matrix $M \in F_2^{n\times N'}$: the first N columns of M are the columns $\begin{pmatrix} [E_1(x)]_1 \\ \vdots \\ [E_1(x)]_n \end{pmatrix}$ for each $x \in \{0,1\}^n$. Then, the next N columns are the same but we replace E_1 with E_2, and similarly with the third and last where we have E_3^{-1} and E_4^{-1} respectively instead of E_1. In other words,

$$M = \left(\begin{pmatrix} [E_1(0)]_1 \\ \vdots \\ [E_1(0)]_n \end{pmatrix} \cdots \begin{pmatrix} [E_2(0)]_1 \\ \vdots \\ [E_2(0)]_n \end{pmatrix} \cdots \begin{pmatrix} [E_3^{-1}(0)]_1 \\ \vdots \\ [E_3^{-1}(0)]_n \end{pmatrix} \cdots \begin{pmatrix} [E_4^{-1}(0)]_1 \\ \vdots \\ [E_4^{-1}(0)]_n \end{pmatrix} \cdots \right).$$

Because E_1, E_2, E_3, E_4 are permutations, the matrix M contains at least 2 different non-zero rows, therefore $\dim(M) \geq 2$. Also,

$$M \cdot \boldsymbol{e}_{LRST} = E_1(L) \oplus E_2(R) \oplus E_3^{-1}(S) \oplus E_4^{-1}(T).$$

We can conclude that $H \subseteq \mathrm{Ker}(M)$ and $\dim(\mathrm{Ker}(M)) = N' - \dim(M) \leq N' - 2$. From this, we can conclude that $\dim(H) \leq N' - 2$. □

Lemma 4. *If $f = \Pi$, we have $\dim(H) = N' - 1$ with overwhelming probability.*

Proof. Let $H_j = \mathrm{span}\{\boldsymbol{e}_{L^i R^i S^i T^i}\}_{i \in [j]}$. We will show that if $\dim(H_j) \leq N' - 2$, then, with constant probability, $\dim(H_{j+1}) = \dim(H_j) + 1$. Let $H_j^\perp$ be the dual of H_j, so

$$x \in H_j \iff \forall y \in H_j^\perp, \langle x, y \rangle = 0.$$

We have $\dim(H_j) + \dim(H_j^\perp) = N'$. This, in particular, implies that $\dim(H_j^\perp) \geq 2$. We can in turn conclude that there exist two distinct non-zero vectors $\boldsymbol{v}_1, \boldsymbol{v}_2 \in H_j^\perp$. This, in turn, implies that there exists $\boldsymbol{v}^* \in H_j^\perp$ such that $|\boldsymbol{v}^*| \leq 2N'/3$. One can indeed easily check that if $|\boldsymbol{v}_1|, |\boldsymbol{v}_2| > 2N'/3$ then $|\boldsymbol{v}_1 + \boldsymbol{v}_2| \leq 2N'/3$.

For a random tuple (L, R, S, T), we then have

$$\begin{aligned}\Pr\left[\boldsymbol{e}_{LRST} \in H_j\right] &\leq \Pr\left[\langle \boldsymbol{e}_{LRST}, \boldsymbol{v}^* \rangle = 0\right] \\ &\leq \left(\frac{1}{3}\right)^4 + 6\left(\frac{1}{3}\right)^2\left(\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^4 = \frac{41}{81}.\end{aligned}$$

This gives $\Pr[\boldsymbol{e}_{LRST} \notin H_j] \geq 40/81$.

However, for $f = \Pi$, the tuples $\{L^i R^i S^i T^i\}_{i \in [1..q]}$ are not entirely random. Indeed, although for tuple $j+1$ the values (L^{j+1}, R^{j+1}) are chosen uniformly at random, the output (S^{j+1}, T^{j+1}) is generated randomly without repetition. For a fixed query, this changes the output distribution by at most $O(j/N^2) = O(1/N)$ (since there are $O(N') = O(N)$ queries in total, so $j \leq O(N)$). We thus obtain, for any j,

$$\Pr\left[\boldsymbol{e}_{L^{j+1} R^{j+1} S^{j+1} T^{j+1}} \notin H_j\right] \geq \frac{40}{81} - O\left(\frac{1}{N}\right).$$

This implies that, with overwhelming probability, $\dim(H_{j+1}) = \dim(H_j) + 1$. Since $q = 4N'$, this then implies that with overwhelming probability $\dim(H_q) \geq N' - 1$. □

We remark that we have not been able to find any improvement to the attack using quantum techniques. In particular, we do not believe that above adversary can benefit from any speed-up in the quantum setting.

C Proof of Theorem 3 (Classical n-Bit Security)

The proof adopts the setup and the reductions as that of Theorem 2. In particular, the steps up to (11) are identical, and we have

$$\mathbf{Adv}_{(\mathrm{QuEME}_{\boldsymbol{K}}^{E,E'})^{\pm};\Pi^{\pm}}(\mathcal{A}) \leq$$
$$\mathbf{Adv}_{(\mathrm{QuEME}^{\boldsymbol{\pi},\tilde{\pi}})^{\pm};f^{\pm}}(\mathcal{A}) + 4 \cdot \mathbf{Adv}_{E}^{\mathrm{sprp}}(\mathcal{A}') + \mathbf{Adv}_{E'}^{\text{ra-sprp}}(\mathcal{A}'') + \frac{q^2}{2^{2n}}. \quad (33)$$

In the remainder, we will focus on the remaining distance between $\mathcal{O} = (\mathrm{QuEME}^{\boldsymbol{\pi},\tilde{\pi}})^{\pm}$ and $\mathcal{P} = f^{\pm}$ and use the H-coefficient technique (Lemma 1) to bound it.

C.1 Additional Notation

As in Section 6.1.2 we assume that $\mathcal{A}$ has unbounded computational power and we measure its complexity only by the number of oracle calls it makes. All queries are recorded in a transcript $\tau = \{(L^i, R^i, S^i, T^i, \mathrm{d}^i) \mid i \in [1..q]\}$, with $\mathrm{d}^i \in \{\text{fwd}, \text{inv}\}$ indicating the query direction. We partition $[1..q]$ into $\mathcal{I}^*$, containing the query indices where both output blocks are *fresh*, and $\mathcal{I}$, containing the query indices where one of the output blocks collides with an earlier block at the same position.

We expand the transcript with $\tau^* = \{(\widehat{L}^i, \widehat{R}^i, X^i\widehat{S}^i, \widehat{T}^i) \mid i \in [1..q]\}$. In the real world $\mathcal{O}$, these are the actual values within the evaluation of $\mathrm{QuEME}^{\boldsymbol{\pi},\tilde{\pi}}$. In the ideal world, these are sampled slightly different from the sampler of Section 6.1.2.

In order to define our sampler, we first define two undirected bipartite graphs G and H. The vertices of G are the q_1 distinct values $L_1, \ldots, L_{q_1}$ in the set $\{L^i \mid i \in [1..q]\}$ and the q_2 distinct values $R_1, \ldots, R_{q_2}$ in the set $\{R^i \mid i \in [1..q]\}$ (we will soon specify how we pick these labels). We put an edge between L_j and R_k if $(L_j, R_k, S, T, \mathrm{d}) \in \tau$ for some S, T, d. The graph H is defined identically, but over the ciphertexts $\{(S^i, T^i) \mid i \in [1..q]\}$ instead of the plaintexts.

Let γ (resp., η) be the number of components in G (resp., H). We label these components $G^{(1)}, \ldots, G^{(\gamma)}$ and $H^{(1)}, \ldots, H^{(\eta)}$. For $t \in [1..\gamma]$ let $q_1^{(t)}$ (resp., $q_2^{(t)}$) be the number of L-nodes (resp., R-nodes) in $G^{(t)}$. Similarly, for $t \in [1..\eta]$ let $q_3^{(t)}$ (resp., $q_4^{(t)}$) be the number of S-nodes (resp., T-nodes) in $H^{(t)}$. Define the cumulative sums

$$Q_b^{(j)} = \sum_{i=1}^{j-1} q_b^{(i)}$$

for each $j \in [1..\gamma]$ when $b \in \{1, 2\}$ and each $j \in [1..\eta]$ when $b \in \{3, 4\}$. We assume the labeling of the L-nodes in G is such that the nodes $\{L_k \mid Q_1^{(j)} + 1 \leq k \leq Q_1^{(j+1)}\}$ are in $G^{(j)}$, and likewise for the R-nodes, S-nodes, and T-nodes.

For simplicity we assume there are no cycle queries; the case when there are cycle queries can be similarly handled. Let $\widehat{L}_1, \ldots, \widehat{L}_{q_1}$, $\widehat{R}_1, \ldots, \widehat{R}_{q_2}$, $\widehat{S}_1, \ldots, \widehat{S}_{q_3}$

and $\widehat{T}_1, \ldots, \widehat{T}_{q_4}$ be the distinct values we need to choose for each permutation. The sampler $\mathcal{S}$ is defined as follows:

1. $\mathcal{S}$ first samples a $(X^1, \ldots, X^q)$ uniformly from the set Λ of all $(X^1, \ldots, X^q)$ satisfying the condition that on any (non-empty) path P of even length in G or H,
$$\bigoplus_{i \in P} X^i \neq 0.$$
This allows us to assign a label $\delta^G_{j,k}$ to each edge (L_j, R_k) in G. This label will be defined $\delta^G_{j,k} := X^i$, where i is such that $L^i = L_j, R^i = R_k$ (such an i must exist for the edge to be part of G). Similarly we assign the label $\delta^H_{j,k} := X^i$ to each edge (S_j, T_k) in H such that $S^i = S_j, T^i = T_k$;
2. Next, $\mathcal{S}$ samples $(\widehat{L}_1, \ldots, \widehat{L}_{q_1}, \widehat{R}_1, \ldots, \widehat{R}_{q_2})$ uniformly from the set Γ^G of all solutions to the q bi-variate equations $\widehat{L}_i \oplus \widehat{R}_j = \delta^G_{i,j}$ satisfying the constraint that $\widehat{L}_1, \ldots, \widehat{L}_{q_1}$ are all distinct and $\widehat{R}_1, \ldots, \widehat{R}_{q_2}$ are all distinct;
3. Finally $\mathcal{S}$ samples $(\widehat{S}_1, \ldots, \widehat{S}_{q_3}, \widehat{T}_1, \ldots, \widehat{T}_{q_4})$ uniformly from the set Γ^H of all solutions to the q bi-variate equations $\widehat{S}_i \oplus \widehat{T}_j = \delta^H_{i,j}$ satisfying the constraint that $\widehat{S}_1, \ldots, \widehat{S}_{q_3}$ are all distinct and $\widehat{T}_1, \ldots, \widehat{T}_{q_4}$ are all distinct.

The sets Λ, Γ^G, and Γ^H will be analyzed further in the next section.

C.2 Analysis of Idealized QuEME

We define the following bad events on the random coins of f and $\mathcal{S}$:

bad_0: For some distinct $i, i' \in [q]$ with $i > i'$:
- $\mathrm{d}^i = \mathrm{fwd}$ and $(S^i, T^i) = (S^{i'}, T^{i'})$; or
- $\mathrm{d}^i = \mathrm{inv}$ and $(L^i, R^i) = (L^{i'}, R^{i'})$;

bad_1: For some distinct $i, i' \in [q]$ with $i > i'$ and $X^i = X^{i'}$:
- $\mathrm{d}^i = \mathrm{fwd}$ and $\left(S^i = S^{i'} \vee T^i = T^{i'}\right)$; or
- $\mathrm{d}^i = \mathrm{inv}$ and $\left(L^i = L^{i'} \vee R^i = R^{i'}\right)$;

bad_2: For some $i \in [q]$:
- $\mathrm{d}^i = \mathrm{fwd}$ and (S^i, T^i) completes a cycle in H; or
- $\mathrm{d}^i = \mathrm{inv}$ and (L^i, R^i) completes a cycle in G.

For bad_2, this event implies that the i^{th} query completes a cycle with nodes coming from the first $i-1$ queries. We define $\mathsf{bad} = \mathsf{bad}_0 \vee \mathsf{bad}_1 \vee \mathsf{bad}_2$.

Bad Transcripts. Recall that the bad events (and hence bad probabilities) are only defined in the ideal world. Analogous to (12), we have

$$\Pr[\mathsf{bad}] \leq \Pr[\mathsf{bad}_0] + \Pr[\mathsf{bad}_1 \mid \neg\mathsf{bad}_0] + \Pr[\mathsf{bad}_2 \mid \neg\mathsf{bad}_0]\ . \tag{34}$$

Now, bad_0 involves a random collision over $2n$ bits, with $\binom{q}{2}$ choices of the two indices i and i'. Thus,

$$\Pr\left[\mathsf{bad}_0\right] \leq \frac{\binom{q}{2}}{N^2} \leq \frac{q^2}{2N^2}. \tag{35}$$

We now bound the probability of bad_1 to happen. Consider a fixed pair of indices $i > i'$ with $\mathrm{d}^i = \mathrm{fwd}$. Then, $S^i, S^{i'}, T^i, T^{i'}, X^i, X^{i'}$ are uniformly random in $[N]$, up to a possible distinctness constraint on X^i and $X^{i'}$. Similarly for $\mathrm{d}^i = \mathrm{inv}$, $L^i, L^{i'}, R^i, R^{i'}, X^i, X^{i'}$ are uniformly random in $[N]$, up to a possible distinctness constraint on X^i and $X^{i'}$. Therefore,

$$\Pr\left[\mathsf{bad}_1 \mid \neg\mathsf{bad}_0\right] \leq \frac{\binom{q}{2} \cdot 2}{N(N-1)} \leq \frac{q^2}{N^2}. \tag{36}$$

Finally, a cycle of length $2m$ (with $m \geq 2$, since a cycle of length 2 would imply bad_0) will need $2m$ collisions for the cycle and give a choice of $2m$ indices and a choice of whether the first node is on the left or the right; since the choice of this "first node" is arbitrary, we divide the total count by m. This gives

$$\begin{aligned}
\Pr\left[\mathsf{bad}_2 \mid \neg\mathsf{bad}_0\right] &\leq \sum_{m\geq 2} \frac{q(q-1)\ldots(q-2m+1)\cdot 2}{N(N-1)\ldots(N-2m+1)\cdot m} \\
&\leq \sum_{m\geq 2} \frac{2q^{2m}}{mN^{2m}} \\
&= \frac{q^2}{N^2} \sum_{m\geq 2} \frac{2}{m} \left(\frac{q^2}{N^2}\right)^{m-1} \\
&\leq \frac{q^2}{N^2} \sum_{m\geq 2} \left(\frac{1}{2}\right)^{m-1} \leq \frac{q^2}{N^2}.
\end{aligned} \tag{37}$$

Substituting (35)-(37) in (34) gives

$$\Pr\left[\mathsf{bad}\right] \leq \frac{2.5q^2}{N^2}. \tag{38}$$

Good Transcripts. Suppose (τ, τ^*) is a good transcript. Let q_1, q_2, q_3, q_4 be the number of distinct values of L^i, R^i, S^i, T^i, respectively, in τ. Further suppose that in τ^*, there are r distinct values of X^i, with the number of queries they appear in being $t_1, \ldots, t_r$, where $t_1 + \ldots + t_r = q$.

In the real world, for each $j \in [4]$, the probability that π_j is compatible with (τ, τ^*) is $1/(N)_{q_j}$, and the probability that $\widetilde{\pi}$ is compatible with (τ, τ^*) is $1/[(N)_{t_1} \cdot \ldots \cdot (N)_{t_r}]$. Thus,

$$\Pr\left[\mathcal{D}_{\mathcal{O}} = (\tau, \tau^*)\right] = \frac{1}{(N)_{q_1} \cdot \ldots \cdot (N)_{q_4} (N)_{t_1} \cdot \ldots \cdot (N)_{t_r}}. \tag{39}$$

In the ideal world, we have q distinct outputs of $2n$-bit random functions, and three independent uniform samples from the sets Λ, Γ^G, and Γ^H, so

$$\Pr[\mathcal{D}_{\mathcal{P}} = (\tau, \tau^*)] = \frac{1}{(N^2)^q |\Lambda||\Gamma^G||\Gamma^H|}. \tag{40}$$

Since we assumed that there are no cycles in G and H, we have $q_1 + q_2 - \gamma = q_3 + q_4 - \eta = q$. Then, from (27) we have

$$|\Gamma^G| \geq \prod_{j=1}^{\gamma} \left[\left(N - Q_1^{(j)}\right)^{q_1^{(j)}} \left(N - Q_2^{(j)}\right)^{q_2^{(j)}} \right] \cdot \frac{1}{N^q}, \tag{41}$$

$$|\Gamma^H| \geq \prod_{j=1}^{\eta} \left[\left(N - Q_3^{(j)}\right)^{q_3^{(j)}} \left(N - Q_4^{(j)}\right)^{q_4^{(j)}} \right] \cdot \frac{1}{N^q}. \tag{42}$$

Similarly, we can show that

$$|\Lambda| \geq N^q \left[\prod_{j=1}^{\gamma} \frac{(N)_{q_1^{(j)}} (N)_{q_2^{(j)}}}{N^{q_1^{(j)} + q_2^{(j)}}} \right] \left[\prod_{j=1}^{\eta} \frac{(N)_{q_3^{(j)}} (N)_{q_4^{(j)}}}{N^{q_3^{(j)} + q_4^{(j)}}} \right]. \tag{43}$$

We observe that

$$\begin{aligned} \left(N - Q_1^{(j)}\right)^{q_1^{(j)}} (N)_{q_1^{(j)}} &= \prod_{k=0}^{q_1^{(j)}-1} \left(N - Q_1^{(j)}\right)(N-k) \\ &\geq \prod_{k=0}^{q_1^{(j)}-1} \left(N - Q_1^{(j)} - k\right) N = \left(N - Q_1^{(j)}\right)_{q_1^{(j)}} N^{q_1^{(j)}}, \end{aligned}$$

so that

$$\prod_{j=1}^{\gamma} \left(N - Q_1^{(j)}\right)^{q_1^{(j)}} (N)_{q_1^{(j)}} \geq (N)_{q_1} N^{q_1}. \tag{44}$$

We can show a similarly that

$$\prod_{j=1}^{\gamma} \left(N - Q_2^{(j)}\right)^{q_2^{(j)}} (N)_{q_2^{(j)}} \geq (N)_{q_2} N^{q_2}, \tag{45}$$

$$\prod_{j=1}^{\eta} \left(N - Q_3^{(j)}\right)^{q_3^{(j)}} (N)_{q_3^{(j)}} \geq (N)_{q_3} N^{q_3}, \tag{46}$$

$$\prod_{j=1}^{\eta} \left(N - Q_4^{(j)}\right)^{q_4^{(j)}} (N)_{q_4^{(j)}} \geq (N)_{q_4} N^{q_4}. \tag{47}$$

From (41)-(47) we can see that

$$\frac{|\Gamma^G||\Gamma^H||\Lambda|}{(N)_{q_1}(N)_{q_2}(N)_{q_3}(N)_{q_4}} \geq \frac{1}{N^q}. \tag{48}$$

From (39), (40) and (48) we get

$$\frac{\Pr\left[\mathcal{D}_{\mathcal{O}} = (\tau, \tau^*)\right]}{\Pr\left[\mathcal{D}_{\mathcal{P}} = (\tau, \tau^*)\right]} \geq \frac{N^q}{(N)_{t_1} \cdot \ldots \cdot (N)_{t_r}} \geq 1. \tag{49}$$

Conclusion. From (38) and (49), using the H-coefficient technique of Lemma 1, we obtain that the remaining advantage in (33) is upper bounded by $2.5q^2/N^2$. This completes the proof, recalling that $N = 2^n$.

D Simulation of Mirror Theory

We perform a small-scale simulation to verify the mirror theory, and concretely the lower bound (27) on the number of solutions to a system of equations of the form (24). The simulation gets as input a random system of equations, first identifies the connected components, and next "solves" the equations in the components according to different strategies.

Identifying Sets of Connected Components. We recall that there is an edge between the variables X_i and Y_j if and only if there is an equation $X_i \oplus Y_j = \delta_{i,j}$. We generate a list of equations sorted by index i and one sorted by index j for retrieving the edges quickly, and then we apply a classic breadth-first traversal of the graph to obtain the components. The procedure is described in Algorithm 4.

Naive Assignment Strategy. One way to generate the solutions is by the following. We initialize two sets of remaining values, to $\{0,1\}^n$: S_X for the variables X and S_Y for the variables Y. We take the largest component, select a value α for its root, and for every other node in the component rule out $\alpha \oplus \Delta$ from the corresponding variable (S_X for X_i and S_Y for Y_j), where Δ is prescribed by the path from the root to the node. We consider the second largest component, select a value β for its root, and again for every other node in the component rule out $\beta \oplus \Delta$, where Δ is the prescribed path from the root to the node. We proceed until there is no component left (thus obtaining a solution) or there is no valid value for a root (thus implying there is no valid solution). This method is not practical as it generates every solution to the system of equations and rounds in doubly exponential time on the size of the input. On the other hand, it seems to be the only approach to give the exact number of solutions.

Approximated Assignment Strategy. We next describe a method to compute an approximation on the number of solutions. Denote the number of components by t and their sizes by $|C^{(j)}|$. We denote by $\Delta_{i,j}$ the set of solutions to the connected components i, j such that the components $C^{(i)}$ and $C^{(j)}$ collide. Then, by the formula of the cardinality of a union, we get

$$2^{nm} - |\text{solutions}| = \sum_{k=1}^{m} (-1)^{k-1} \sum_{\{i_1,j_1\}>\cdots>\{i_k,j_k\}} |\Delta_{i_1,j_1} \cap \cdots \cap \Delta_{i_k,j_k}|,$$

Algorithm 4 Identifying components

Input: list of equations of the form $X_i \oplus Y_j = \delta_{i,j}$
Output: list of connected components

Sort the equations $X_i \oplus Y_j = \delta_{i,j}$ by i, and store in L_X
▷ There needs to be a place to mark the indices i.
Sort the equations $X_i \oplus Y_j = \delta_{i,j}$ by j, and store in L_Y
▷ There needs to be a place to mark the indices j.
for all i **do**
    Start a stack with the element $(X, i, 0)$
        ▷ Elements of the pile are of the form (Z, l, Δ) with $Z = X$ or Y to indicate the target list L_X or L_Y, l the corresponding index in the target list and Δ the difference with the stating element, also called the root.
    Start a stack for recording the elements of the current connected component with the root element $(X, i, 0)$
    while the stack is not empty **do**
        Pop the first element off the stack (Z, l, Δ)
        if l is not marked in the list L_Z **then**
            for all $X_i \oplus Y_j = \delta_{i,j}$ in L_Z with $i = l$ (if $X = Z$) **do**
                Add $(Y, j, \Delta \oplus \delta_{i,j})$ to the pile and to the component
            end for
            for all $X_i \oplus Y_j = \delta_{i,j}$ in L_Z with $j = l$ (if $Y = Z$) **do**
                Add $(X, i, \Delta \oplus \delta_{i,j})$ to the pile and to the component
            end for
            Mark l in the list L_Z
        end if
    end while
    Register the list if it contains more than one element
        ▷ This connected component has either no elements or is an isolated point.
end for
Return list of connected components

where $>$ is any ordering.

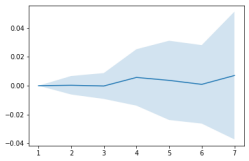
A first observation is that for all $\{i_1, j_1\} \neq \{i_2, j_2\}$,

$$|\Delta_{i_1,j_1} \cap \Delta_{i_2,j_2}| \times 2^{nm} = |\Delta_{i_1,j_1}| \times |\Delta_{i_2,j_2}|,$$

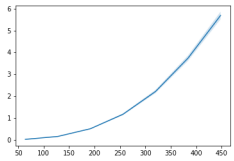
as the difference between the value of the roots of $C^{(i_1)}$ and $C^{(j_1)}$ and between $C^{(i_2)}$ and $C^{(j_2)}$ are independent. Indeed, if $\{i_1, j_1\} \cap \{i_2, j_2\} = \emptyset$, the sets are independent, whereas if $\{i_1, j_1\} \cap \{i_2, j_2\} \neq \emptyset$, we consider the differences of the root values which are independent as we are in a vector space.

Then the computation of the different terms depends on whether the different sets $\{i_1, j_1\} > \cdots > \{i_k, j_k\}$ have an intersection or not. In that direction, for a set $\{i_1, j_1\} > \cdots > \{i_k, j_k\}$, we define the graph $G_{\{i_1,j_1\}>\cdots>\{i_k,j_k\}}$ with vertices $1, \ldots, m$ and an edge between vertices a and b if and only if there exists an l such that $\{i_l, j_l\} = \{a, b\}$. For a graph G on vertices $1, \ldots, m$, we define

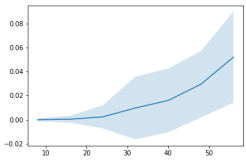
$$S_G = \frac{1}{2^{nm}} \sum_{G' \cong G} \Big| \bigcap_{i,j \in G'} \Delta_{i,j} \Big|,$$



(a) exact approach ($n = 5$)



(b) first approximation ($n = 11$)



(c) second approximation ($n = 8$)

Fig. 5: Simulation results, depicting the difference of the logarithm of the results between simulation and conjectural prediction by number of equations. The line is the mean over 100 trials, the surrounding area is the variability.

where $\cong$ denotes graph isomorphism. We let C_G be the number of connected components of G and P_G the number of non-isolated points.

The observation extends to the computation of any S_G, where G has no cycle, and to unions of non-connected sub-graphs. The value S_G can be bounded by $(\sum_i |C_i|^2)^{P_G}/2^{n(m-C_G)}$. This means that this method of approximation is very well-suited for cases where the value $\sum_j |C^{(j)}|^2$ is controlled. For random systems, $\sum_j |C^{(j)}|^2 = O(q)$. Then, a first approximation can be made by taking

$$\frac{|\text{solutions}|}{2^{nm}} = \prod_{i,j} \left(1 - \frac{|\Delta_{i,j}|}{2^n}\right) + O\left(\frac{q^3}{2^{2n}}\right).$$

More advanced approximations can be made by considering cycles of successively bigger sizes. For example, by considering the cycles of size 3, we get a better approximation with an error in $O(q^4/2^{3n})$.

Results. We performed a simulation with different approximations, namely the exact approach for $n = 5$, a first approximation for $n = 11$, and a better approximation for $n = 8$. The simulation took 10 hours on an Intel i5-6500U CPU, and the results are depicted in Figure 5. The results support the mirror theory lower bound (27) for small values of n. Unfortunately, expanding the analysis to larger values of n becomes quickly infeasible.

E Proof of Lemma 2 (Quantum to Classical Security)

Fix an adversary $\mathcal{A}_{QQ}$. Its qPRP advantage is equivalently described in Game1 below.

Game1 → Game2. We transform Game1 into Game2 by prepending two random permutations to f.

Game1: prp-game($\mathcal{A}_{QQ}$)	Game2: permutation blinding
$b \xleftarrow{\$} \{0,1\}$	$b \xleftarrow{\$} \{0,1\}$
If $b = 0$	**If** $b = 0$
$\boldsymbol{\pi} = \pi_1, \pi_2, \pi_3, \pi_4 \xleftarrow{\$} \text{perm}(n)$	$\boldsymbol{\pi} = \pi_1, \pi_2, \pi_3, \pi_4 \xleftarrow{\$} \text{perm}(n)$
$f = \text{QuEME}^{\boldsymbol{\pi}, E'}$	$f = \text{QuEME}^{\boldsymbol{\pi}, E'}$
If $b = 1$	**If** $b = 1$
$f \xleftarrow{\$} \text{perm}(2n)$	$f \xleftarrow{\$} \text{perm}(2n)$
	$h_L, h_R \xleftarrow{\$} \text{perm}(n)$
	$f' := f \circ (h_L \| h_R)$
$b' \leftarrow \mathcal{A}_{QQ}^{f}(\cdot)$	$b' \leftarrow \mathcal{A}_{QQ}^{f'}(\cdot)$
Win if $b = b'$	**Win if** $b = b'$

As $\text{QuEME}^{\boldsymbol{\pi}, E'}$ already starts with two random permutations $\pi_1 \| \pi_2$ in parallel, adding an extra layer of permutations does not change its distribution. In the ideal world, the addition of the extra layer of permutations does not change the distribution either, and hence

$$\Pr\left[\mathcal{A}_{QQ} \text{ wins Game1}\right] = \Pr\left[\mathcal{A}_{QQ} \text{ wins Game2}\right].$$

Game2 → Game3. We now replace the two freshly added permutations with random small range functions distributed according to $S_n(r)$, which is defined as follows.

Definition 1. *$S_n(r)$ is a distribution on functions from $\{0,1\}^n$ to $\{0,1\}^n$ sampled as follows:*

- *Draw a random function g from $\{0,1\}^n \to [r]$;*
- *Draw a random injective function h from $[r] \to \{0,1\}^n$;*
- *Output the composition $h \circ g$.*

Notice that any function f drawn from $S_n(r)$ satisfies $|Im(f)| \leq r$.

The transition from Game2 to Game3 is described below.

Game2: permutation blinding	Game3: small range functions
$b \xleftarrow{\$} \{0,1\}$	$b \xleftarrow{\$} \{0,1\}$
If $b = 0$	**If** $b = 0$
$\boldsymbol{\pi} = \pi_1, \pi_2, \pi_3, \pi_4 \xleftarrow{\$} \text{perm}(n)$	$\boldsymbol{\pi} = \pi_1, \pi_2, \pi_3, \pi_4 \xleftarrow{\$} \text{perm}(n)$
$f = \text{QuEME}^{\boldsymbol{\pi}, E'}$	$f = \text{QuEME}^{\boldsymbol{\pi}, E'}$
If $b = 1$	**If** $b = 1$
$f \xleftarrow{\$} \text{perm}(2n)$	$f \xleftarrow{\$} \text{perm}(2n)$
$h_L, h_R \xleftarrow{\$} \text{perm}(n)$	$h_L, h_R \xleftarrow{\$} S_n(r)$
$f' := f \circ (h_L \| h_R)$	$f' := f \circ (h_L \| h_R)$
$b' \leftarrow \mathcal{A}_{QQ}^{f'}(\cdot)$	$b' \leftarrow \mathcal{A}_{QQ}^{f'}(\cdot)$
Win if $b = b'$	**Win if** $b = b'$

Zhandry [62] proved that a small range function behaves like a random permutation up to a certain bound.

Lemma 5 (Zhandry [62]). *For any r, for any quantum adversary $\mathcal{A}$ performing q quantum queries, we have $\mathbf{Adv}_f^{\text{q-prp}}(\mathcal{A}) = O\left(\frac{q^3}{r}\right)$, for $f \xleftarrow{\$} S_n(r)$.*

From Lemma 5, we subsequently obtain

$$\Pr\left[\mathcal{A}_{QQ} \text{ wins Game2}\right] \leq \Pr\left[\mathcal{A}_{QQ} \text{ wins Game3}\right] + O\left(\frac{q^3}{r}\right).$$

Game3: Classical Emulation. Consider the following adversary $\mathcal{A}'_{QC}$ against Game1 with classical queries:

- Pick $h_L, h_R \xleftarrow{\$} S_n(r)$. Let Z_L, Z_R be the ranges of h_L, h_R respectively, so they are each subsets of $\{0,1\}^n$ of size r;
- Define $f' := f \circ (h_L \| h_R)$;
- Query $f(x,y)$ for each $(x,y) \in Z_L \times Z_R$ for a total of r^2 queries. From these queries, recover the truth table of f';
- Emulate the quantum circuit $\mathcal{A}_{QQ}^{f'}(\cdot)$ and output $b' \leftarrow \mathcal{A}_{QQ}^{f'}(\cdot)$.

By definition, $\mathcal{A}'_{QC}$ outputs exactly the same output as $\mathcal{A}_{QQ}^{f'}$, and hence

$$\Pr\left[\mathcal{A}'_{QC} \text{ wins Game1}\right] = \Pr\left[\mathcal{A}_{QQ} \text{ wins Game3}\right].$$

Moreover, $\mathcal{A}'_{QC}$ is a quantum algorithm that performs r^2 queries to f. We remark that $\mathcal{A}'_{QC}$ does not a priori know the sets Z_L, Z_R but it can reconstruct them with $\widetilde{O}(r)$ queries to h_L and h_R. Then, it can recover the whole truth table of f on the input set $Z_L \times Z_R$, hence he knows the full truth table of f'. From there, it can emulate the quantum queries to f' using its truth table, which can be done efficiently, assuming efficient Quantum RAM.

Conclusion. We can now conclude:

$$\begin{aligned}
\mathbf{Adv}^{\text{q-prp}}_{\text{QuEME}^{\boldsymbol{\pi},E'}}(\mathcal{A}_{QQ}) &= \Pr\left[\mathcal{A}_{QQ} \text{ wins Game1}\right] \\
&\leq \Pr\left[\mathcal{A}_{QQ} \text{ wins Game3}\right] + O\left(\frac{q^3}{r}\right) \\
&= \Pr\left[\mathcal{A}'_{QC} \text{ wins Game1}\right] + O\left(\frac{q^3}{r}\right) \\
&= \mathbf{Adv}^{\text{prp}}_{\text{QuEME}^{\boldsymbol{\pi},E'}}(\mathcal{A}'_{QC}) + O\left(\frac{q^3}{r}\right).
\end{aligned}$$

F AES Specification and Known Attacks

The internal state of AES [18] is 128 bits. The three standardized versions have a key of size 128, 192, or 256 bits, and internally evaluate 10, 12, or 14 rounds,

respectively. Note that, given Grover's algorithm, AES-256 would be able to reach key recovery security of 128 bits, but when used in most common modes, collisions on internal states could provide other kind of attacks, potentially better than classical attacks under some assumptions on the attackers.

Specification of AES. We provide a basic description of AES-128 and we point to [17] for more details. The state of AES-128 is composed of elements of $\mathbb{F}_{256}$, organized in a 4×4 matrix:

$$\begin{bmatrix} \alpha_0 & \alpha_4 & \alpha_8 & \alpha_{12} \\ \alpha_1 & \alpha_5 & \alpha_9 & \alpha_{13} \\ \alpha_2 & \alpha_6 & \alpha_{10} & \alpha_{14} \\ \alpha_3 & \alpha_7 & \alpha_{11} & \alpha_{15} \end{bmatrix}.$$

AES-128 is composed of 10 rounds, which internally consist of four operations:

- AddKey, which XORs the state with the round key (see below);
- SubBytes, which applies the AES S-box on all individual elements α_i;
- ShiftRows, which shifts the i^{th} row by i positions;
- MixColumns, which multiplies each column by a fixed matrix.

The last round omits the MixColumns operation and applies one extra AddKey.

The round keys are derived from the 128-bit master key K as follows. First, write $K = (k_0 \| k_1 \| k_2 \| k_3)$. Then, the first round key K_0 equals K, and round keys K_i for $i = 1, \ldots, 10$ are defined as $K_i = (k_{4i+4} \| k_{4i+5} \| k_{4i+6} \| k_{4i+7})$, where

$$\begin{array}{l} k_{4i+4} = \text{SubWord}\left(\text{RotWord}(k_{4i+3})\right) \oplus k_{4i} \oplus rc_i \\ k_{4i+5} = k_{4i+4} \oplus k_{4i+1} \\ k_{4i+6} = k_{4i+5} \oplus k_{4i+2} \\ k_{4i+7} = k_{4i+6} \oplus k_{4i+3} \end{array},$$

and

$$rc_i = \begin{pmatrix} X^i \bmod X^8 + X^4 + X^3 + X + 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Best Known Attacks on AES-128. We list the best known attacks on AES-128 in the secret key setting in Table 3 and in other settings in Table 4.

Table 3: Currently known cryptanalysis for round-reduced AES-128 in the secret-key model.

Attack	Rounds	Time	Data	Reference
Mixture Differential	5	$2^{21.5}$	$2^{21.5}$	[4]
Yoyo	5	2^{33}	$2^{13.3}$	[57]
Partial Sum	5	2^{40}	2^{8}	[60]
Rectangle	5	2^{23}	2^{9}	[22]
Rectangle	5	$2^{16.5}$	2^{15}	[22]
Improved Square	5	2^{35}	2^{33}	[25]
Boomeyong	5	2^{49}	2^{49}	[56]
Rectangle	6	2^{80}	2^{26}	[22]
Partial Sum	6	2^{44}	$2^{34.5}$	[25]
Truncated Differential	6	$2^{78.7}$	$2^{71.3}$	[3]
Boomeyong	6	$2^{79.72}$	$2^{79.72}$	[56]
Impossible Differential	7	$2^{117.2}$	$2^{112.2}$	[46]
Meet-in-the-Middle	7	2^{116}	2^{116}	[23]
Impossible Differential	7	2^{113}	$2^{105.1}$	[9]
Impossible Differential	7	$2^{110.9}$	$2^{104.9}$	[45]
Meet-in-the-Middle	7	2^{99}	2^{97}	[20]

Table 4: Currently known cryptanalysis for round-reduced AES-128 in the related-key/chosen-key/known-key model.

Attack	Rounds	Time	Data	Reference
Related-key				
RK Boomerang	7	2^{97}	2^{97}	[7]
Chosen-key				
Multi-collision	9	2^{55}	2^{55}	[26]
Multiple-of-n	9	2^{64}	2^{64}	[28]
Known-key				
Uniform Distribution	10	2^{64}	2^{64}	[27]
Uniform Distribution	12	2^{82}	2^{82}	[29]